**The London School of Economics and Political Science**

# Distributed Development of Large-Scale Distributed Systems: The Case of the Particle Physics Grid

**Avgousta Kyriakidou-Zacharoudiou**

A thesis submitted to the Department of Management, Information Systems and Innovation Group, of the London School of Economics for the degree of Doctor of Philosophy, London

May 2011

# DECLARATION

I certify that the thesis I have presented for examination for the PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

# ABSTRACT

Developing a Grid within High Energy Physics for the Large Hadron Collider particle accelerator is characterised as a highly collaborative, distributed and dynamic systems development effort. This research examines the way this distributed Grid is developed, deployed and provided as a service to the thousands of physicists analysing data from the Large Hadron Collider. The particle physics community has always been at the forefront of computing with a tradition of working in large distributed collaborations, therefore providing a "distinctive" case of distributed systems development practice. The focus of concern is the collaborative systems development practices employed by particle physicists in their attempt to develop a usable Grid.

The research aims to offer lessons and practical recommendations to those involved in globally distributed systems development and to inform the information systems development literature. Global software development presents unaddressed challenges to organisations and it is argued that there is an urgent need for new systems development practices and strategies to be created that can facilitate and embrace the rapid changes of the environment and the complexities involved in such projects. The contribution of the study, therefore, is a framework of guidance towards engendering what the author defines as "Hybrid Experimental Agile Distributed Systems Development Communities" revealing a set of dynamic collaborative practices for those organisational contexts engaged in distributed systems development. The framework will allow them to reflect on their own practice and perhaps foster a similarly dynamic flexible community in order to manage their global software development effort.

The research is in the form of an interpretative qualitative exploratory case study, which draws upon Activity Theory, and frames the Grid's distributed development activity as a complex overarching networked activity system influenced by the context, the community's tools, rules, norms, culture, history, past experiences, shared visions and collaborative way of working. Tensions and contradictions throughout the development of this Grid are explored and surfaced, with the research focusing on how these are resolved in order for the activity system to achieve stability. Such stability leads to the construction of new knowledge and learning and the formation of new systems development practices. In studying this, practices are considered as an emergent property linked to improvisation, bricolage and dynamic competences that unfold as large-scale projects evolve.

*To my parents*

# ACKNOWLEDGEMENTS

This thesis is dedicated to my beloved mum and dad. You brought me into this world, you have been there when I needed you most and you are everything one could ask for from their parents. I am really proud that you are my parents and I want everyone to know that. Mum, I love you and I want you to know that your sacrifices have not been in vain. Dad, we have been discussing my PhD from day 1. I will never forget how happy you were when I announced to you that some day I would become a "Dr.". Now that this wonderful journey is coming closer to an end, your absence is even worse. You are not with us anymore and I would like to take this opportunity to tell you that you are the one who made this happen. You were always giving me the strength to carry on because you were a fighter. You are a true inspiration, you are one in a million and I was lucky enough to have you as my dad. I love you and I miss you and I hope that you are proud of me.

I could not have done this PhD without the love and support of my wonderful husband, Ioanni Zacharoudiou. You made this possible. Thank you for all the sleepless nights, thank you for the long and endless discussions about my work, thank you for looking after me, thank you for everything. I am also grateful to my sisters, Evi and Maria, who supported me, kept me happy and for not letting me lose my enthusiasm. Their patience, love and understanding have been invaluable. I would like to extend my thanks to all my family in Cyprus (and it is a big family!) and my in-laws, for their continuous encouragement and support whilst completing this dissertation and for not letting my priorities waver. Kypro, Evi, Denni, Nana and Avgousta, you are like brothers and sisters to me. I am blessed to have you in my life.

The Information systems and Innovation Group at the LSE has been a wonderful academic environment since my days as an MSc student and immediately afterwards as a PhD student and graduate teaching assistant. I will always be grateful to my supervisor Dr. Will Venters. Dear Will, thank you for choosing me to work with you, thank you for believing in me, thank you for your commitment and all your support throughout this challenging period of my PhD. I could never describe with words how grateful I feel, but I am honoured to have you as a supervisor and as a friend. A big thank you also to your dear wife Angela and my beautiful bridesmaid, your daughter, Hannah. I remember once you told me that you wanted this PhD to be a great thesis that you could show to your future PhD students with pride. I hope that, as your first PhD student, I have succeeded in this. I hope that this is a thesis that makes you and will always make you proud.

I also like to thank my second supervisor Dr. Tony Cornford for his precious advice during the early writings on this research. Dear Tony, you make people want to improve themselves. Thank you for your guidance and your confidence in my academic abilities. Special thanks also to Dr. Steve Smithson for reading a draft of this thesis and providing valuable comments as well as to Dr. Carsten Sørensen for his support. Also thank you to Dr. Jannis Kallinikos and Dr. Susan Scott who acted as my examiners in the MPhil upgrade process. Their constructive initial comments have helped shape this dissertation.

To all my friends and colleagues who I encountered at LSE, especially Maha Shaikh, Katerina Voutsina, Dionysis Demetis, Vasiliki Baka, Eleni Lioliou, Ioanna Chini and Yingqin Zheng, I thank you very much for your support and for the invaluable advice you have offered me. A special thanks to Ritsa Pitta, Isidora Kourti and Marios Tziannaros who have been great friends and have always been there for me.

Last but not least the wider particle physics community at the UK and CERN deserve a special mention. I am grateful to all the people I interviewed and observed as they have been an integral part of this thesis. Thank you very much for inviting me to your community and making me feel a part of it. I have been following you around the world these past few years and it has been an amazing intellectual journey. Thank you for allowing me to study your Grid development effort, and for your openness and support throughout this research.

## TABLE OF CONTENTS

## List of Figures

## List of Tables

# 1 Introduction – Research issues

## 1.1 Problem domain and scope of the research

This research is a critical study of collaborative work practices within global systems development activities. It particularly examines how large-scale distributed systems are developed collaboratively in a globally distributed manner, through a case study of a Grid development project, the Large Hadron Collider computing Grid (LCG) project, in particle physics (PP). The aim of this thesis is to observe how the development of the Grid unfolds in a large, complex, distributed community and track events and activities over a period of time, aiming at understanding such events in their natural setting and capturing the subjective experiences of the community's members during the development process. This research examines the development of the Grid, as an open-ended unfolding process. Clearly the technical tasks required in achieving this are significant; however this research's interest lies in the collaborative processes and practices employed in developing such Grid technology as a socio-technical achievement, rather than as a purely technical artefact.

The focus of this study is a globally distributed community, the PP community, who are required to collaborate in a virtual and distributed way in order to develop tools to facilitate their physics research. Particle physicists await a new particle accelerator, the Large Hadron Collider (LHC), to fully begin its operation to collide protons to recreate the conditions that prevailed in the universe at the earliest moments of "Big Bang", in a search for the "Higgs boson" particle (Doyle 2005). These interactions produce 15 perabytes of data annually and thousands of physicists all around the world are eager to analyse them (Lloyd 2006). The storage and analysis of the data requires more than 100 000 central processing units (CPUs) of processing power and a huge amount of space. A traditional approach would be to centralise this capacity at one location, however, in the case of the LHC, various external pressures (such as funding) mean a novel globally distributed model is needed. Their approach to this is the development of a Grid, a form of globally distributed computing system constituted of diverse software and hardware resources. The mission of this large widely distributed community is to build and maintain the "world's biggest Grid". The LCG project was therefore built around this effort in an attempt to bring physicists from all around the world together to develop, deploy and maintain this Grid (Britton, Burke et al. 2006a).

The systems development setting studied here, the PP community, is a community with a tradition of solving their own technological problems. As such, they often find themselves in the roles of developers as well as users. This community is well-known for the development of other cutting edge distributed systems (notably the World Wide Web) and is itself highly distributed and virtual with distinctive culture and tradition (Knorr-Cetina 1999), so presenting a context where distinctive collaborative practices emerge. The more narrow area the research focuses upon is the development, deployment of Grid middleware and applications and user support.

This study particularly explores the collaborative systems development practices of the PP community in their attempt to develop a usable Grid and the collaboration formed around this development, with the aim to offer answers into the wider context of global distributed systems development. This collaborative development is explored through three distinctive perspectives: 1) *structure* and in particular how this whole effort is organised and managed as well as how competitive relationships are balanced, 2) *history/culture* and particularly how their distinctive and "exceptional" history and tradition influence and facilitate their current practice, and 3) *process*, focusing more on the processes and practices employed when developing and deploying Grid middleware components and applications to run on top of the middleware. In studying these, practices are considered as an emergent property linked to improvisation, bricolage (Ciborra 2002) and dynamic competences which unfold as large-scale projects evolve.

Exploring this case, the researcher argues that the development of Grids poses new and under-explored opportunities for understanding collaborative distributed systems development. Grids are a new form of large-scale systems that are highly distributed, both in conception/construction and operation (Foster and Kesselman 2003). Grid computing promises to distribute and share computing resources "on tap" and provide transparent communication and collaboration between virtual groups (Bremer, Campolargo et al. 2002). The PP Grid is part of an initiative, the e-Science vision (Hey and Trefethen 2005), which aims to produce not just a working system but a new generation of computing technology that may potentially have significant impact on scientific research, and perhaps foreshadow the “next generation Internet” (Carr 2005). Yet, developing and implementing such a complex information infrastructure requires collaboration among a range of dispersed groups, and flexibility and adaptability to volatile requirements (Berman, Geoffrey et al. 2003). Developing a Grid is indeed argued to be a significant systems development challenge, because it must be done as a distributed virtual collaborative effort. It involves coordinating the actions of a huge range of people of different culture, education and skill, all working within different institutions around the globe. Vital in this enterprise are the ability to share knowledge about the project, and to support innovating new technology and new work practices. In part this is achieved by careful attention to developing a strong sense of community among participants, and constructing a range of repositories of information about the project, the people, and the Grid itself.

The Grid is technically complex. Yet the complexity of the globally distributed systems development effort within the LCG project sometimes “exceeds” the Grid’s technical challenges. It is the mobilisation of resources, the creation of alliances, the drawing of boundaries, the “struggle” between order and chaos and experimentation and discipline, which constitute the fluidity and complexity in the emergence of the Grid infrastructure.

The following research questions are thus addressed in the context of the PP Grid development project providing lessons for the wider context of e-Science and global software development (GSD): *(1) What is the nature of systems development for such a large-scale global project? (2) What practices emerge in response to the demands of such development? (3) How do these practices influence the development of large-scale and global systems? (4) What lessons are learned from the development of the PP Grid that can be translated into the wider field of GSD?*

## 1.2 Research approach

This research is undertaken as a part of a wider study, the Pegasus project, which explores the working practices of particle physicists. The Pegasus project’s focus is broader, particularly focusing on the usability of the Grid and mainly looking at the

UK Grid for particle physics (GridPP) project (a PP country-focused project contributing to the wider LCG). This study focuses uniquely on the systems development practices of particle physicists developing the Grid middleware. It particularly looks at the wider LCG project by encompassing elements from GridPP and from the Enabling Grids for e-Science (EGEE) project (an EU funded project jointly developing middleware with LCG) as contributors to LCG.

The researcher participated in all the activities and meetings of the Pegasus project, from its conception till its completion and played a key role in the project's success. Being part of this project benefited the researcher greatly and made this study better as it provided easier and greater access to participants, financial independence and the ability to discuss the findings and analysis within the wider project and with a steering group who provided help and direction to the wider project. The researcher conducted the work in collaboration with the remaining team. The researcher's input to the project is further discussed in the methodology chapter.

The research is in the form of an interpretative qualitative exploratory case study (Myers 1997) that addresses the research questions by focusing on 1) how collaboration occurs, 2) how new knowledge and expertise are created and shared, 3) the importance of trust and how competition is facilitated and 4) on the work practices and technologies that support this distributed collaborative endeavour. The case study strategy allows for a thorough understanding of the Grid distributed development context and offers deep insights into the complexities inherent in the development process. The central arguments of this thesis are thus derived from the comprehensive analysis of the data collected from the empirical case study.

In this research a range of theoretical concepts with a specific focus on collaborative practice (Schatzki, Knorr-Cetina et al. 2001) and communities (Wenger 1998; Porra 1999) are employed. The theoretical framework is drawn from activity theory (AT) (Engestrom 1990), which encompasses such concepts, and frames the development effort as a complex overarching activity system consisting of networked interacting sub-activity systems, which share the same object. The development activity is influenced by the context, the community's rules, norms, culture, history, past experiences, shared visions and collaborative practices (Nardi 1996). The Grid's development is understood as a series of contradictions between the elements of this overarching activity system, which are in a continuous process of getting resolved in order for the activity system to achieve stability and balance. Contradictions are considered to be the major source of dynamism and development in AT, since based on the emerging problems and conflicts, people have to re-consider their position and collectively re-construct their shared understanding, knowledge and practices (Bertelsen 2003). Tensions and frictions throughout the development of this Grid are thus explored and surfaced, focusing on how these are resolved in order for the overarching activity system to achieve stability, leading to the construction of new knowledge and learning and the formation of new systems development practices.

AT is considered a relevant theoretical framework to explain collaboration – and hence global systems development activity – as a complex work activity influenced by the context, the community, mediated artefacts, rules and division of labour of subjects undertaking this activity (Korpela, Mursu et al. 2002). It allows for an in-depth examination on how the Grid is collaboratively constructed in order to fulfil the objects of a global community and of the learning activities taking place throughout the process.

This research offers an opportunity to apply a theory little known beyond Scandinavian literature to an information systems (IS) research project and therefore inform mainstream IS research regarding the theory's advantages. However, to the author's knowledge, this research represents one of the first attempts to apply the theory to the study of globally distributed systems development of large-scale distributed systems. One of the theoretical contributions of this study is to the methodology of AT which does not reflect well the complexity of such globally distributed collaborative activities. This study suggests that future developments of AT should be based on what the author defines as an "AT meta-framework 2.5" reflecting on ideas of the third generation AT but based on a recursive development of the second generation AT in which network accounts are undertaken. This study therefore informs and enriches AT in order to better reflect such widely distributed collaborative activities undertaken within the context of complex communities such as the one identified in this study.

## 1.3 Motivation of the research, contributions and relevance of the study

***Motivation***

The work starts with the assumption that systems development beyond the smallest scale is inherently a collaborative activity (Whitehead 2007). With the current trend of globalisation where the information technology (IT) industry is becoming more and more globally interconnected (Sengupta, Chandra et al. 2006), the management, development and maintenance of software has increasingly become a political, multi-site, multi-cultural distributed undertaking, undertaken on a larger and more diffused scale as virtualised efforts distributed in time and space (Herbsleb, Paulish et al. 2005).

Today there are more software projects running in geographically distributed environments and the so-called GSD is becoming a norm in the software industry (Damian and Moitra 2006). Systems development now involves knowledge from multiple domains that a single developer (or organisation) does not possess. Collaboration then becomes a necessity because of the need to achieve levels of collective intelligence (Ye 2005). Organisations have to cope with the increasing internationalisation of business forcing collaboration and knowledge sharing simultaneously and thus there is now a need for new ways of thinking about how global collaborative development should be fostered and how knowledge should be shared in distributed groups (Gammelgaard 2010). It is not surprising that literature stresses that such collaborative and virtual systems development endeavours blur the boundaries between sectors and disciplines and have consequences for the development practices involved in the construction of systems (Agerfalk, Fitzgerald et al. 2009). With the globalisation of the development process (Sengupta, Chandra et al. 2006) and the new forms of open technologies such as Grids, the limitations of traditional systems development practices (and even contemporary agile practices) have become obvious (Fitzgerald 2000; Parnas 2006).

While software projects across multiple sites are created and carried out, with GSD becoming a business necessity for various reasons such as reduction of costs, scarcity of resources and the need to locate development closer to customers, pertinent questions that emerge and are still unaddressed are “How are successful global projects carried out? How much progress have we made towards equipping the software industry with necessary techniques or tools to overcome the difficulties faced by global teams?" (Damian 2003). It is therefore argued that there is an urgent need for case studies which can facilitate a deep understanding of the current challenges and ones which can contribute solutions, effective strategies and practices, and lessons learned on how to address problems of technical, social and cultural nature in GSD (Smite, Wohlin et al. 2010).

GSD is argued to demand new, different development practices, since the nature of the problem and the environment are different (Damian and Moitra 2006). It is suggested that there is a need to develop alternative perspectives on distributed systems development, ones that put forward improvisation and emergent change as fundamental aspects of systems development in organisations (Bansler and Havn 2004). Herbsleb and Moitra (2001) indeed claim that there is still "significant understanding to be achieved, methods and techniques to be developed and practices to be evolved before this GSD phenomenon becomes a mature discipline and its benefits are fully understood". Russo and Stolterman (2000) and Fitzgerald (2000) claim that new systems development practices, or guidelines for successful systems development, should be drawn from "best practice" development situations that prevail today. They further suggest, researchers should focus on examining systems development practice in real-world situations, and in real-world development projects.

This research is thus motivated by the debate around GSD and attempts to bridge this evident "gap" by providing suggestions, lessons learned and practical recommendations from the PP context that will fill in the urgent need for new and more efficient systems development practices. This research rejects the idea of "best-practice", instead it explores a case that is interesting and unusual in the way they develop systems and therefore provides juxtaposition to more orthodox accounts with lessons emerging from reflection on both.

***Contributions***

The contribution of the study, therefore, is a framework of guidance towards engendering, managing and maintaining what the researcher defines as *"Hybrid Experimental Agile Distributed Systems Development Communities*" (HEAD_SDC) revealing a set of dynamic collaborative practices relevant for those organisational contexts engaged in distributed systems development, which will allow them to reflect on their own practice and perhaps foster a similarly dynamic agile community in order to manage their GSD effort. HEAD_SDC, exhibit similar but also additional and “distinctive” characteristics to the ones already defined in traditional communities of practice (CoP) (Wenger 1998). Although CoP are spontaneously emerging groups (Wenger and Snyder 2000) and therefore it is difficult to structure their spontaneity and provide the enabling conditions – a vision, motivation, the systems and structures – to support their development and to sustain their activities, the community identified in this study has been developed and sustained through collaborative means of working by the whole PP collaboration; therefore, providing lessons based on what particle physicists believe can engender and maintain such a collaborative virtual community. This HEAD_SDC is identified as a new form of collaboration which is highly distributed, clustered, organic, collaborative and democratic, it is experimental and agile with tremendous computing expertise and not following pre-determined plans, methods and structures.

The HEAD_SDC framework of guidance provides the practical contribution of the thesis to GSD and agile development literatures, since it reveals multiple dimensions of agility that are applicable to GSD contexts. However, it also represents this study’s theoretical contribution to theories of collaboration and communities (Weick and Roberts 1993; Wenger 1998; Porra 1999). Reflecting on AT and the collaborative practices from the case material, HEAD_SDC allow for a renewed understanding of collaborative work within a different form of a widely distributed community. Literature on theories of communities, is centred around the debate whether traditional CoP (Kimble and Hildreth 2005), colonial systems (Porra and Parks 2006) and a collective mind (D'Eredita, Misiolek et al. 2005; D'Eredita and Chau 2005), which are considered a vital aspiration for global organisations (Rios, Aguilera et al. 2009), can emerge in a widely distributed and virtual context. This framework, therefore, informs and enriches such literature by identifying distinctive characteristics of such a global virtual community, which reflect on the realistic challenges and opportunities of performing GSD in such a distributed environment.

***Relevance of the study***

This study is based on the belief that lessons of this community's distributed collaborative development of a Grid can provide insights for other fields with less experience of engaging with and developing large-scale technologies in a globally distributed way. This however, then poses the further challenge to assess to what extent the ways of working of this community, which are embedded in the "doing" of PP and within its complicated and unique context of massive physical apparatus, colossal data volumes and long traditions of large collaborative teams, can inform wider systems development efforts both within e-Science and beyond? If there are indeed lessons to be drawn from projects such as the LCG, it is here argued that it is necessary to look beyond the simple replication of a technological infrastructure within another domain. Instead, the Grid should be considered as a socio-technical information infrastructure such that the context within which LCG practices and accomplishments exist may be better understood. From this position and with a perspective that establishes a Grid as a reflection and construction of its context, valuable insights can be gained.

This dissertation thus differentiates itself from previous studies focused on GSD projects such as open source or outsourcing related projects, as it explores a "unique" and, as claimed by Chompalov, Genuth et al. (2002), an "exceptional", collaborative, democratic and highly technologically competent distributed community in their attempt to develop a Grid to meet their LHC computing needs. The technology under study is obscure with distinctive characteristics and it is deployed under very specific circumstances such as the tight LHC deadline and the need to meet the particle physicists' requirements and expectations, something which is expected to influence the systems development practices employed. While some of the PP collaborative practices reflect elements of those employed in open source and agile development, PP is a unique organisation exhibiting various distinctive characteristics that can offer lessons to GSD.

The PP Grid and the LCG project in particular is indeed an appropriate area for the study of large-scale distributed development for a number of reasons. Firstly, the organisational context of LCG, reflecting that of PP (Traweek 1988; Knorr-Cetina 1999), is highly distributed with the LCG being spread across 170 computing centres in 34 countries. Their Grid, which has been under development since 2001 is considered to be one of the world's largest e-infrastructures (Britton, Burke et al. 2006b). Secondly, it is being developed within a community which has for many years worked as a globally distributed collaboration that thrives on democratic debates and discussions and has shown their ability to work pragmatically and creatively under extreme pressures and tight deadlines (Lloyd 2006).

Developing this Grid is a highly collaborative, distributed and dynamic systems development effort but one that also presents challenges in technical, organisational and social terms. It is indeed an example of large-scale systems development on a global scale and one that seems to demonstrate a spirit of agility, improvisation and emergent change. No prediction or prescription is intended. Rather, the findings of this thesis aim to provide guidance to other organisational contexts that attempt to perform distributed systems development of Grids and other technologies within their work.

### 1.4 Structure of the thesis

This thesis is structured as follows:

*Chapter 2* builds the argument of the study by reviewing literature from fields relevant to distributed software development, which has studied aspects of such software processes before. Literature on theories of collaboration and communities (such as CoP, colonial systems, collective mind) is also reviewed in an attempt to enrich the community element of the AT framework employed and provide the basis for framing the contribution of the research. Finally, Grid literature is reviewed in order to define what the Grid is for this study. This literature is used to develop a coherent stance for the research.

*Chapter 3* presents AT that gives perspective to this study. An overview of the theory – discussion of its structures and principles – is undertaken to demonstrate its suitability to the study. The theory's conceptual tools applied in this research, such as the activity triangle, the structure of an activity, the concept of contradictions and the model of expansive learning, are further discussed in detail.

*Chapter 4* describes different traditions in research in order to situate this study within the field of interpretative IS research. The chosen case study approach is outlined and the way in which the research was conducted and data were analysed is demonstrated.

*Chapter 5* describes the case – the wider organisational context within which the Grid "emerges". The three main projects involved in the Grid's development under study are presented and details on their structure, objectives and existing culture are discussed. This chapter complements Chapter 6, providing an overview of the different elements which when combined make up this case.

*Chapter 6* provides a detailed presentation of the relevant data obtained from the empirical study of the Grid's distributed systems development. The focus here is on how the Grid is developed collaboratively by a number of distributed actors. This chapter is split into two important parts. The first part provides details on how the collaboration around the development is structured, how the whole effort is organised, how knowledge is socialised and how the extensive communication flow helps coordinate the work. The second part of this chapter presents details on how the development work is performed, therefore, provides descriptions of the development, deployment and user support process.

*Chapter 7* thoroughly analyses and synthesises the data obtained from the case to shed light into how this large-scale Grid is developed collaboratively in a global way and how new systems development practices emerge to facilitate such development. The AT's conceptual tools are used to structure this analysis in an attempt to break the data down and elucidate the intricacies and contradictions involved in the overarching "distributed systems development activity system". Through the community's effort to resolve such "contradictions", the researcher unveils evidence of their expansive learning and discusses the lessons learned and the new practices that emerged as the activity system achieved its balance. The thesis' adoption of the second generation AT led to a new recursive structure for distributed collaborative activities, which contributes towards the third generation of AT and presents one of the theoretical contributions of this study.

*Chapter 8* takes a step further to propose a framework for creating HEAD_SDC. In this chapter, the findings and analysis are extensively discussed shedding light and facilitating the development of the above framework. The framework's ideas are examined and compared to the already existing literature on communities (discussed in Chapter 2). The relevance of this new framework – which is developed, based on the PP community's lessons learned – to GSD is also portrayed. This framework is argued as a contribution to the discourse of GSD and agile development and the theories of collaboration and communities.

*Chapter 9* concludes this dissertation. Firstly, a thesis synopsis is provided after which the outcomes of the research questions identified in the introduction chapter are briefly discussed. The major contribution of the research to both theory and practice is then presented followed by the limitations, challenges and ideas to stimulate further research in the areas covered by this study.

Figure 1.1 provides the broader structure of the thesis.

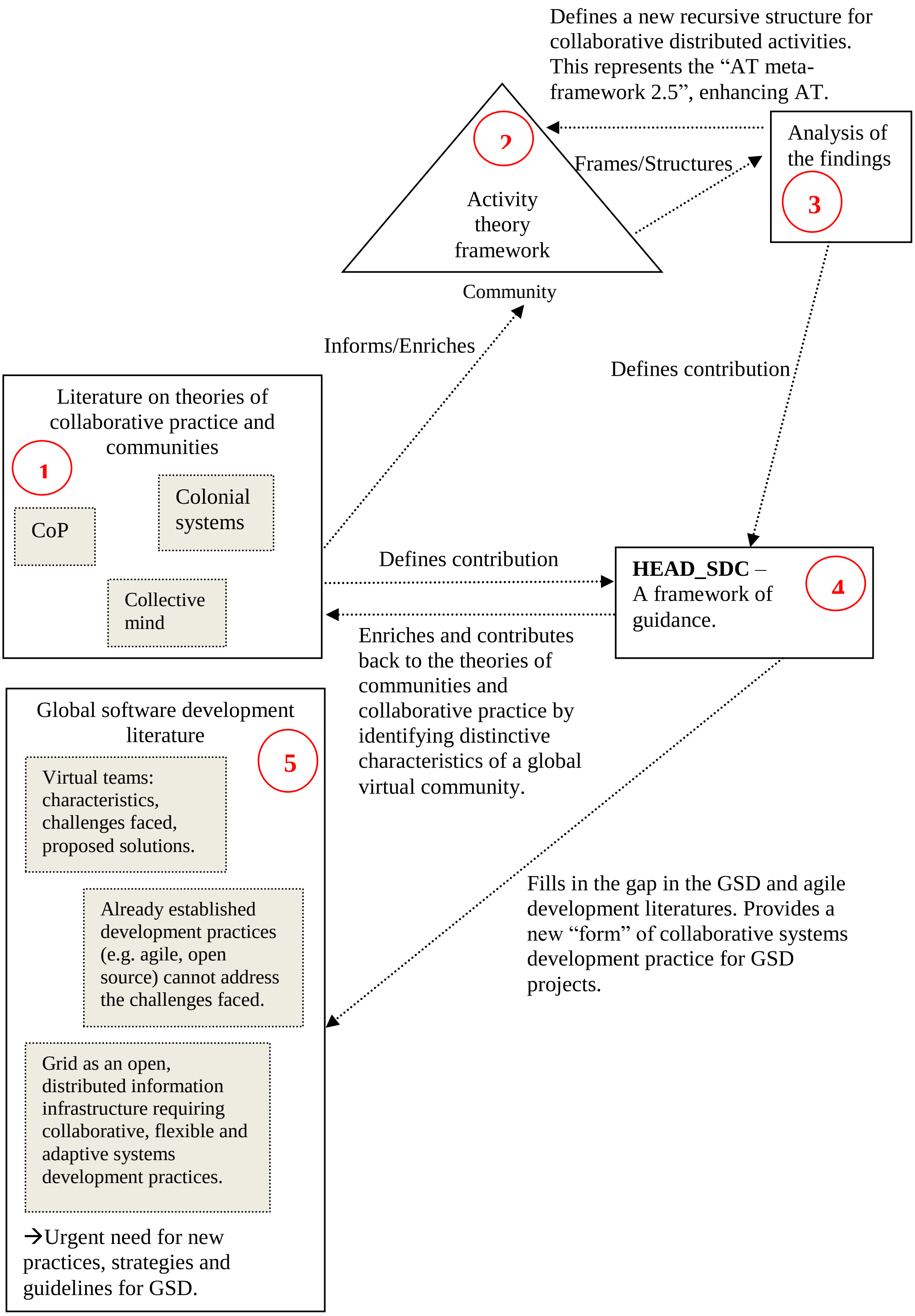


Figure 1.1 PhD roadmap

# 2 Literature review

## 2.1 Introduction

This chapter presents a thorough review of the relevant literature in accordance to the objectives of this study. The chapter consists of three basic parts. Part 1 reviews literature that draws upon debates and literature from fields such as GSD, open source development and global outsourcing that are familiar with collaborative distributed software development processes and which previously have studied aspects, issues, practices and challenges of such software processes. The lack of evidence in the literature to address the challenges faced by GSD projects leads to a thorough discussion and review of relevant work on whether already proposed systems development practices, such as traditional methodologies, contemporary agile development approaches and open source approaches are appropriate to deal with these challenges. In light of this the perceived "gap" in the information systems development (ISD) and GSD literature is presented, urgently needing new and effective strategies and practices that can address problems of technical, social and cultural nature in GSD.

Part 2 of the chapter reviews literature on collaborative practice and communities and in particular it focuses on concepts such as CoP achieving shared identity and learning (Lave and Wenger 1991; Wenger 1998), Weick's concept of collective mind (Weick and Roberts 1993) and Porra's (1999) work on colonial systems. Such literature is reviewed in an attempt to enrich AT and particularly its concept of community, which does not reflect well the community being studied here. This enables a better analysis of the research findings. Such literature also helps define the contribution of this study, as the framework of guidance developed here draws upon and extends the concepts of CoP, colonial systems and collective mind in a global virtual development setting, thereby defining a new "distinctive" global hybrid systems development community.

Finally, Part 3 introduces the Grid and conceptualises it as a large-scale e-information infrastructure, demanding a distributed collaborative systems development effort.

The chapter is structured as follows: Section 2.2 sets out the scene for GSD focusing on the characteristics of such software processes, while Section 2.2.1 discusses virtual teams in GSD, perceived challenges and proposed solutions. Section 2.2.2 provides details on existing systems development practices (focusing on agile practices and open source development practices) that are perceived as the "silver bullet" to global systems development. Section 2.2.3 follows which identifies the "gap" in the literature and the urgent need for new development paradigms or guidelines that can inform future progress in distributed development after which the second part of the literature focusing on collective practice and communities is presented in Section 2.3. Section 2.4 presents details on the Grid and Section 2.5 summarises this chapter.

## 2.2 Part 1: Global – Virtual software development

ISD is at the core of the IS field (Fitzgerald, Russo et al. 2002). It is considered to be the fundamental process – a collaborative problem solving and intellectually complex activity (Vessey and Glass 1998) – carried out when engaging with technology to achieve a specific purpose in a specific context.

With the current trend of globalisation and the problem of turbulent business environments, ongoing innovations in information communication technologies (ICT) have made it possible to cooperate in a distributed fashion (Herbsleb, Paulish et al. 2005). The continuing advances and progress in ICT as well as in bandwidth and capacity and the use of advanced applications have turned the IT industry toward globally distributed software development in an attempt to find the silver bullet (Brooks 1987) of high-quality software delivered cheaply and quickly (Agerfalk and Fitzgerald 2006). From originally quite small co-located projects, enabled by technological advances, companies now embark on major complex software development projects running in geographically distributed environments (Oshri, Kotlarsky et al. 2008) and GSD is therefore "becoming a norm in the software industry" (Damian and Moitra 2006).

GSD is argued to be performed in a highly uncertain and complex environment, consisting of widely dispersed groups of people with limited physical interaction (Herbsleb and Mockus 2003). Development in such a context needs to be quickly adaptable to the rapidly changing requirements and that is why communication, coordination and speed are all seen to be important aspects of the GSD process (Herbsleb, Mockus et al. 2001). A number of characteristics seem to define GSD projects with Smite and Borzovs (2006) suggesting the most important as being multi-sourcing, geographic distribution, temporal and socio-cultural diversity, linguistic and contextual diversity as well as political and legislative diversity.

It is argued that GSD is increasingly becoming common practice in the software industry, mostly because of its ability to develop software at remote sites, ignoring the geographical distance and benefit from access to a qualified resource pool and reduced development costs (Herbsleb 2007). One of the advantages of GSD is that it provides opportunities for developers in dispersed locations to communicate, collaborate as well as build and share knowledge repositories (Che and Zhao 2005). It is indeed claimed by Herbsleb and Moitra (2001) that a number of factors have accelerated this new trend with the most profound being: 1) the need to capitalise on the global resource pool to successfully and cost-competitively use scarce resources, wherever located; 2) the business advantage of proximity to the market, including knowledge of customers and local conditions 3) the quick formation of virtual teams to exploit market opportunities; 4) the severe pressure to improve time-to-market by using time-zone differences in "round-the-clock" development; and 5) the need for flexibility to capitalise on merger and acquisition opportunities wherever they present themselves. One-Ki, Probir et al. (2006) suggest that the open source development context is very similar to the GSD environment as it is characterised by a distributed environment, collaborative and rapid development among virtual teams and rapid evolution as the environment changes. Similarly, the globalisation's effects on outsourcing of software production, have made outsourcing take up global dimensions and thereby become an international complex undertaking which requires a tremendous amount of support and interaction (Yalaho 2006).

Today's complex, competitive and dynamic business environment indeed requires adaptive, flexible and responsive organisations. The access to a larger pool of expertise compels companies to form globally virtual alliances with other organisations in order to survive and thus switch to GSD or offshore outsourcing of software products and services (Kotlarsky, Van Fenema et al. 2008).

Virtual alliances are defined as a network organisation consisting of independent enterprises (organisations, groups, individuals) that come together to explore an opportunity, business or market (Kasper-Fuehrer and Ashkanasy 2001). They can take various forms and structures with some being more permanent, interactive, knowledge intensive or complex than others (Panteli and Sockalingam 2005). Burn, Marschall et al. (2002) have created a virtual inter-organisational arrangement model which emphasises the strategic relationships between organisations including power, dominance and collaboration. Their model introduces three types of virtual alliances, the *star alliance,* the *value alliance* and finally the *co-alliance,* with the third model experiencing the highest levels of task conflicts as the nature of work is unstructured, but also experiencing a sense of mutuality and belonging which can lead to commitment and improve cooperation.

In management science, inter-organisational alliances have been investigated for a long time (Jha and Watson-Manheim 2007) and the term "Virtual Organisations" has been in common usage since at least 1992 (Davidow and Malone 1992; Jha and Watson-Manheim 2007). A "virtual organisation" (VO), it is claimed, is an extension of the traditional physical and structural bounded groups that are enabled and cooperate through technological advances, which revolutionise communication between them by establishing linkages for shared knowledge (Panteli and Chiasson 2008). Schultze and Orlikowski (2001) explain that "global electronic workspaces or information devices…together make up the unseen and sprawling empires of VOs", with Panteli and Chiasson (2008) arguing that the advancement of ICT has been related not only to the emergence of virtual society but also to the development of the *"virtual empire"*, which is supposed to have an enormous impact on how people work, communicate and share knowledge. The location of work and our co-workers is now considered irrelevant. It is not surprising therefore that virtuality has been linked with globalisation that is about the "death of distance". However, as Van Binsbergen (1998) puts it: "globalisation is not about the absence or dissolution of boundaries, but about…the opening up of new spaces and new times within new boundaries that we hitherto unconceivable".

Gibson and Gibbs (2006) have tried to identify virtuality as a multi-faced construct which consists of four characteristics: geographical dispersion, electronic dependence, dynamic structural arrangements and national diversity. While there is a diverse set of meanings attached to virtuality, authors agree that being so complex and dynamic in nature, virtuality requires collective and collaborative efforts in order to better appreciate the broader, social, cultural, geographical and technological characteristics that surround it (Webster 2005).

Having defined what GSD and VOs mean to this study, the section below provides a review of the literature on virtual teams, their unique characteristics, the various challenges they face as a part of GSD projects as well as the already proposed suggestions to resolve such challenges. Such topics are reviewed in an attempt to inform this study and identify any relevant “gaps” in the literature.

### 2.2.1 Virtual teams in GSD: Characteristics, challenges, proposed solutions

***Characteristics***

As companies expand globally, face increasing time compression in product development, and use more foreign based subcontracting labour, “virtual teams promise the flexibility, responsiveness, lower costs, and improved resource utilisation necessary to meet ever-changing task requirements in highly turbulent and dynamic global business environments” (Javenpaa and Leidner 1999). Schmidt, Temple et al (2008) argue there are a number of types of teams that fall along a continuum from traditional face-to-face to completely fully virtual and distributed. As they claim, a wide range of teams are left in the middle of this continuum and exhibit a mixed mode of interaction and are therefore called “*hybrid teams”.* Hybrid teams are thus part of a complex spectrum of possibilities between completely virtual and completely traditional (Crowston, Howison et al. 2005). In order to identify the degree of virtuality in a team, Chudoba, Wynn et al. (2005) have introduced three dimensions of virtuality, the team distribution, workplace mobility and variety of practices. However, while several authors associate virtuality and its degree with geographic dispersion e.g. (Walther and Bunz 2005) critics argue that teams may well be highly virtualised when not operating over dispersed borders (Panteli and Chiasson 2008). For example, the usage of ICT not only enables individuals or groups to collaborate synchronously and asynchronously, but it also makes a team virtual in different extents. For a virtual team to be fully distributed, therefore, key factors include the absence of face-to-face interaction but the presence of interaction through digital technologies between team members (Griffith, Sawyer et al. 2003). For a team to be hybrid, the presence of non-digital and digital technologies as well as the presence of face-to-face interaction needs to exist (ibid). Finally, the absence of digital technologies in communication and interaction is a key factor for making the team traditional. For the purpose of this research, virtual teams are seen as hybrid in order to inform the virtual community identified in this study, which has facets of a hybrid community thus portraying similar characteristics. However, this study looks beyond this concept as the identified virtual community also exhibits characteristics of CoP and colonial systems further discussed in Part 2 of this chapter.

Global virtual teams can rapidly form, reorganise, and dissolve when the needs of a dynamic marketplace change, and consist of individuals with differing competencies who are located across time, space and cultures (ibid). They represent a new organisational form that has emerged in conjunction with the globalisation of the socioeconomic and development process (Oshri, Van Fenema et al. 2008).

GSD projects (e.g. open source, global outsourcing) consist of global virtual teams where groups of developers, widely dispersed, work together on joint projects or common tasks through the use of technologies to achieve a shared purpose (Edwards and Sridhar 2002). Within these virtual teams developers collaborate, communicate, share resources, and access remote equipment, through virtual environments, which are a set of online tools, systems and processes interoperating to facilitate the whole development process as well as the collaboration among participants (OSIgroup 2006). Fulk and DeSanctis (1995) have identified various characteristics of collaborative technologies which offer crucial features that benefit the organisations and virtual teams. These include the speed of communication, the increased communication bandwidth, the dramatic reduction in costs of communication, the expanded connectivity with people and machines throughout the globe and the integration of communication with computing technologies. The most commonly suggested collaborative technologies are email, chat (e.g. msn), teleconferencing, videoconferencing, intranet, group calendar, discussion lists, electronic meeting systems, wikis, blogs, etc. (Herbsleb and Mockus 2003). Such collaborative technologies, ranging from electronic systems and videoconferencing, even traditional technologies e.g. telephone or email, play a critical role in supporting virtualised work environments. Recent research also suggests integrating collaborative technologies into integrated development environments in order to offer solutions that deal with breakdowns in communication among developers in virtual software teams (Zhua, Gonga et al. 2007).

Clearly, virtual teams are neither effective simply because of technology nor as a result of organisations wanting to extend their boundaries, but also and most importantly because individuals are able to trust and thus interact and work together in these electronic and non- traditional environments (Panteli and Chiasson 2008a).

***Challenges***

The substantial area of enquiry into remote collaborations has been the subject of a range of research strands over the years with some of the key issues been nicely summarised by Olson and Olson (2000) in their paper "Distance Matters". They argue that distance does matter, it is "not only alive and well, it is in several essential respects immortal" (Olson and Olson 2000). Although distributed systems development with global virtual teams is considered to be a new paradigm in developing large-scale systems (Damian and Moitra 2006), there are still challenges and complexities involved in managing the development, coordinating dispersed team members, sharing knowledge, etc. that need to be addressed for having successful GSD (Herbsleb and Moitra 2001).

Remote activities engender problems that directly confront individuals or groups engaged in these activities, such as ensuring the appropriate means of negotiating interdependencies, issues related to conflicting goals and priorities, articulation of tasks, etc. (ibid). Such challenges of distributed work between virtual teams have for a long time been discussed. The most common challenges that literature in GSD generally raise, stem from geographical dispersion and thus from distance and time-zone differences (Carmel 1999). These lead to coordination and communication breakdowns as people might experience problems with technologies that cannot substitute for face-to-face meetings as they lack interactivity, and time-zone differences reduce opportunities for real-time collaboration (Carmel and Tjia 2005). It is argued that these issues are even more complex in outsourcing arrangements, as the fear of loss of intellectual property, or other proprietary information about products leads to restricted or filtered communication, often seriously impairing this critical channel (Yalaho, Chunling et al. 2004).

Other studies show that distributed work usually worsens the chance for misunderstanding (Van Fenema 2002), there is a lack of trust (Panteli and Sockalingam 2005), asymmetrical distribution of information and knowledge among distributed sites (Carmel 1999; Panteli and Sockalingam 2005) and difficulty in collaborating due to different skills and training as well as mismatches in technologies. Strategic issues, such as deciding how to divide up the work, as well as technical issues such as incompatible data formats and different versions of the same tools, are also argued by Herbsleb and Moitra (2001) to be adding a further barrier to GSD.

GSD is situated within a complex, multi-leveled, multi-site socio-cultural context and it is argued that the environment of global virtual teams is not independent of the local setting and the cultural context (Sarker and Sahay 2004). Huang and Trauth (2007) have identified a number of key challenges in GSD with the most predominant indeed being the cultural diversity and cross-cultural management with Tanner (2009) similarly supporting that communication and culture are key issues that need to be addressed. Oshri, Van Fenema et al. (2008) similarly argue that virtual teams face major challenges in transferring knowledge across remote sites because of cultural differences that may include different languages and national traditions, communication habits, implicit assumptions as well as different values and norms of behaviour. DeSouza and Evaristo (2004) further suggest that dispersed members often adopt unique local routines for working, training and learning which create difficulties in standardising the work practices and might hinder the development of shared understanding and knowledge across the VO. Differences in skills, expertise, technical infrastructure, development tools and methodologies further raise the barriers for knowledge transfer between distributed sites (DeSouza 2003).

Indeed, while co-located teams may develop various memory systems that support knowledge transfer, virtual dispersed teams often face challenges in developing such memory systems that may provide support to where expertise is located and for the transfer of contextual and embedded knowledge (Oshri, Van Fenema et al. 2008). Such challenges exist mostly because distributed teams often experience changes in membership and their distribution decreases communication and increases the possibility for conflicts, misunderstanding and breakdowns (Armstrong and Cole 1995). While in co-located software development teams, joint training and face-to-face meetings facilitate the development of shared understanding and of the feeling of belonging to a group, in GSD this presents a problem (ibid).

***Proposed solutions to GSD challenges***

A number of studies have sought to propose solutions for overcoming the perceived challenges of GSD. Studies have suggested a clear division of labour and tasks between the dispersed sites (Battin, Crocker et al. 2001), inter-site coordination through division of labour which minimises cross-site communication and synchronisation (Mockus and Weiss 2001) as well as the use of technologies that support collaboration in a distributed environment (Cheng, DeSouza et al. 2004). On the other hand, Lee-Kelley, Crossman et al. (2004) have stressed that better performance in virtual teams is achieved through face-to-face meetings as a mechanism for team development and group identity. Qureshi and Zigurs (2001) similarly, play down the role of sophisticated technologies for successful virtualisation and emphasise the need for occasional face-to-face meetings for durable online collaboration. Robey, Khoo et al. (2000) also recommend that face-to-face meetings establish a greater social connection among members with Maznevski and Chudoba (2000) however claiming that the frequency of face-to-face meetings can gradually be reduced with the emergence of task clarity.

Recent research by Kotlarsky and Van Fenema (2008) has created a framework with a set of guidelines on how to eliminate constraints such as distance and time zone differences in distributed collaborative work. They suggest that (1) members of virtual distributed teams should be selected with experience to participate in distributed collaboration, (2) managers of such VOs should invest in the development and maintenance of knowledge and finally they should try to (3) minimise the loss of expertise and experience in organisations by reducing turnover and aiming for long-term employment. Based on a similar perspective Oshri, Van Fenema et al. (2008) argue that creating transactive memory systems, which are combined individual memory systems with communications/transactions between individuals, may enhance specialisation and division of labour and therefore minimise the constraints of sharing knowledge in distributed work, while increasing the awareness of who knows what and improve team performance. Kotlarsky, Van Fenema et al. (2008) however, take a different approach to this and stress the importance of coordination mechanisms, such as *organisation design mechanism, work-based mechanisms, technology-based mechanisms* and *social mechanisms* and technologies in eliminating the challenges of distributed work, especially those attributed to knowledge sharing and social aspects of global projects. While coordination mechanisms for managing knowledge process are important, Oshri, Kotlarsky et al. (2008b) argue that "successful software development efforts depend on timely and accurate coordination of expertise". They propose a hybrid approach in managing and sharing of expertise which encourages the exploitation of expertise within globally distributed projects and yet explores the development and integration of expertise from external sources of knowledge.

While previous research, as discussed above, recommends coordination mechanisms and communication patterns, in order to address the challenges of distributed work, a number of authors suggest the formation of tightly coupled groups that require frequent coordination and synchronisation to be performed within one site (Mockus and Weiss 2001; Herbsleb and Mockus 2003). Therefore, division of labour should be based on geographical location. Contributions to this have proposed the component-based development approach for facilitating distributed development (e.g. (Repenning, Ioannidou et al. 2001), where components can be developed independently from remote locations with minimum inter-site communication and coordination activities, something which minimises the constraints faced on distributed work. However, Kotlarsky, Oshri et al. (2008) argue, that such an approach may in fact result in fewer opportunities to reuse components because of team members' limited exposure to knowledge and therefore they suggest that division of work should be based on technical or functional expertise as this enables virtual reams to utilise the knowledge and expertise of their colleagues regardless of their geographical location.

On a similar perspective, a new stream of literature attempting to provide strategies for successful GSD suggests among others a clear distribution rationale, a sense of teamness, temporary co-location, instant feedback, informing and monitoring practices, appropriate training, human communication and effective tool bases as key factors enabling and facilitating successful global systems development (Paasivaara and Lassenius 2003; Prikladnicki, Audy et al. 2003; Paasivaara and Lassenius 2004; Lings, Lundell et al. 2006; Gratton and Erickson 2007). Olson, Zimmerman et al. (2008) also propose five major clusters of components that are seen as important to the success of global collaborations including the nature of work, the amount of common ground among participants, their readiness to collaborate, their management style and leadership and technology readiness.

Literature on GSD, as already discussed, has focused on solutions to problems related to the geographical dispersion of work. While so far the main focus of various literature on GSD has been on technical aspects related to such projects, for example the development of proper collaborative technologies, e.g. replicated databases, bug-tracking tools and document management systems (Carmel and Agarwal 2001; Herbsleb, Atkins et al. 2002; Smith and Blanck 2002) or the proper application of technical mechanisms and project procedures (Majchrzak, Rice et al. 2000; Herbsleb, Atkins et al. 2002), there is a new stream of literature in the IS field which highlights the importance of social, human and collaborative aspects in dispersed projects (Oshri, Van Fenema et al. 2008). It has been argued that distributed projects, being mostly large-scale and with a number of constraints "demand" social, cultural and collaborative forces (Yan 2008). Formalism and structure cannot hold them together, rather they need to be bound together though shared values and norms, through strong collaborative relationships, shared goals and shared understanding (ibid). Various factors that are seen to contribute towards successful collaborative distributed systems development, such as formal and informal socialisation (Child 2001), trust (Javenpaa, Knoll et al. 1998; Javenpaa and Leidner 1999), motivation and social ties (Ahuja and Galvin 2003) as well as how these can be achieved will now be discussed.

***Successful collaboration, socialization and the formation of trust in GSD***

There is not a clear pattern for creating successful collaborations in GSD in the literature. However, it is possible to distinguish a range of conditions that improve the likelihood of success. Mattessich, Murray-Close et al. (2001) point out that some of the factors necessary for successful GSD collaborations are mutual understanding and respect, informal and personal relationships, open and frequent communication, shared vision, concrete and achievable goals, flexibility and adaptability, and a favourable political and social climate. Cohen and Bailey (1997) particularly stress that shared expectations of work practices, patterns of interactions and communication as well as on how technologies should be used are especially critical in the virtual environment where use of media is integral in accomplishing work activities. Qureshi and Zigurs (2001) on the other hand argue that management motivation has a direct effect on virtual collaboration, rather than technologies which should be seen as devises and not drivers of the collaboration in a GSD project.

While various literature stress factors that can enable and facilitate successful collaboration in GSD, Paasivaara and Lassenius (2003), claim that organisations easily underestimate the need for specific collaborative practices and processes when running global development projects. As they suggest, companies jump into global projects without first planning how to work together with their partners. The existence of only few collaborative practices suitable for GSD complicates things even more. Oshri, Kotlarsky et al. (2007) recommend that better socialisation needs to be achieved in a global development setting in order to help better planning of the distributed work as well as facilitate the creation of collaborative relations and the creation and sharing of collective knowledge. As they argue, the creation of collective knowledge is more likely in organised groups where individuals share a degree of continuity of existence and identity and where a body of common traditions, customs and habits are developed (ibid). Building a sense of collective knowledge ensures effective communication within the collective boundary and means the development of a collective mind (Weick and Roberts 1993) through participation in tasks and social rituals (Orlikowski 2002). However, for this knowledge transfer scenario to be effectively used in distributed contexts, socialisation and frequent interaction which allow for the creation of a community based on shared norms and understanding is needed (Oshri, Kotlarsky et al. 2008).

Through socialisation, the norms, identity and cohesion between team members develop, enabling them to effectively communicate and perform. In distributed contexts, socialisation can be enhanced through technology (Ahuja and Galvin 2003), however, as argued by Oshri, Kotlarsky et al. (2008), its creation and maintenance may require distributed teams to reacquire norms and re-socialise as the project progresses. Indeed, evidence from numerous studies seem to indicate that the creation of commitment and of relationships enabled by periodic face-to-face meetings and socialisation, form an important ingredient for the durability of virtual team-working (Nandhakumar and Baskerville 2006).

Another key aspect enabling successful collaboration in GSD is argued to be the creation of trust (Javenpaa and Leidner 1999). Socialisation and effective knowledge sharing have indeed been proved to facilitate the creation of a trustworthy environment (Brede-Moe and Šmite 2008). Trust has been found to be positively related to global teams' performance but in global environments it develops based on more "identifiable actions such as timely information sharing, appropriate responses to electronic communications, and keeping commitments to virtual team-mates" rather than based on social and emotional attachments (Furst, Reeves et al. 2004).

Weems-Landingham (2008) suggests that in order for global collaborations to be effective, managers should use trust where possible to build bonds and manage personal emotions. Such trust can effectively resolve conflict resolution between global virtual teams and can facilitate the control of the various benefits of conflict, one of which is valuable innovation (Panteli and Sockalingam 2005). Yet, trust is merely a reason to believe that critical resources will be available and committed to interdependent performance in GSD (O'Leary, Orlikowski et al. 2002). Although control mechanisms are viewed as an alternative to trust, these controls (e.g. organisational-based norms, cultures, rules etc.) have proven ineffective within global virtual contexts (Weems-Landingham 2008). O'Leary, Orlikowski et al. (2002) claim that "neither trust, nor control is sufficient for distributed work and they should not be perceived as diametric opposites but as complements existing along the same continuum".

Therefore, other social factors or tactics need to be taken into account and enlisted in their place for having successful distributed collaborative relationships. Tactics here are defined as “attempts to influence other (e.g. expert human resources) to feel, think or behave in a desired fashion” (Weems-Landingham 2008). Tactics that facilitate the development and utilisation of positive instrumental relationships – relationships that are not grounded in the physical but in the psychological proximity – should therefore be evoked, as they promote the strongest bonds between team members.

Establishing team identity/organisational identification for example has been argued to increase affiliation and membership since a sense of belonging to the organisation is created which links distributed members together (Yan 2008). Team identity is defined as the acceptance of interdependent goals and collective commitment and is seen to moderate the often difficult social relations in VOs such as the fragile trust, the easy occurrence of conflicts and the inappropriate attribution as well as to reduce uncertainty and complexity (Hinds and Mortensen 2005; Yan 2008). Although identity is seen as important for having successful global collaborations, tactics that promote such team identity in distributed contexts have not yet been established through systematic research (Weems-Landingham 2008). It is therefore argued that additional research must be carried out to determine tactics which for example promote trust, team identity, etc. that can enhance the virtual team members’ abilities to collaborate effectively in global contexts (ibid).

The section below discusses a number of system development approaches that have been suggested as appropriate solutions to the challenges of GSD, nevertheless their appropriateness is still to be examined.

### 2.2.2 Systems development practices and their suitability in GSD

GSD is claimed to be one of the mega-trends shaping the industry today (Simons 2006). The changing nature of the environment and the “faster metabolism” of business today, require organisations to act more effectively in shorter time frames and to develop software at Internet speed (Ramesh, Cao et al. 2006).

It is claimed that the ISD context with most systems development methodologies, no longer makes sense (Fitzgerald 2000). Today's dynamic environment has given rise to organisations that continuously adapt their strategies, structures and policies to suit this new global environment (Nerur, Mahapatra et al. 2005). However, the traditional plan-driven methodologies lack the flexibility to dynamically adjust the development process. Traditional methodologies are now being questioned because they are rooted in practices and concepts that were relevant to completely different "organisational and technical realities" (Fitzgerald 1994). The first methodologies were initially designed to support the early ISD which was mostly about in-house development of isolated systems from scratch (Avison and Fitzgerald 2003b) and were based on assumptions that IS are closed, stand-alone systems developed within a hierarchical structure and used within closed organisational limits. Nonetheless, the challenges of today's distributed environment, call for new ISD paradigms and practices that are sufficiently "light", recognise the particular character of work in such turbulent environments and can facilitate continuous development, wholeness, integration of different and complex parts, alignment of IS with business objectives and friendliness towards the various actors involved (Rupino da Cuhna and Dias de Figueiredo 2001).

The distribution of companies has become a necessity not only because of the need of shifting labour, but also on pursuit of talented people regardless of location, time zone, etc. that are otherwise unavailable (Ye 2005). Particular attention is hence being given on the opportunities and difficulties associated with sharing knowledge and transferring "best practices" within and across organisations (Orlikowski 2002). Knowledge-based collaboration within systems development has therefore become very important (ibid). Global outsourcing literature suggests that distributed systems development practices that focus on collaborative practices in systems development are important and timely (Lacity and Willcocks 2001; Yalaho 2006) because true collaborative working and trust are fundamental characteristics of successful GSD. Over the years the approaches and practices involved in systems development have undergone refinement and new more effective practices that focus on the collaborative factor have emerged such as agile practices/agility (Nerur, Mahapatra et al. 2005). Currently the software industry is facing two paradigms that are claimed to promise a revolution in software development (Theunissen, Boake et al. 2005). On the one hand, agile software development has emerged as a fast-paced, nimble means of developing software for customers; while on the other hand, there has been an increased adoption of open source software (OSS) development, in the hope of maximising reuse and reducing costs (ibid). These two paradigms offer important knowledge for this study as they reflect a new collaborative form of systems development. It is therefore important to understand them as this study reflects elements of them. The two paradigms are described below.

***Agile approaches***

Agile methods are a relatively recent phenomenon, most famously formally advocated in "the agile manifesto" (Fowler and Highsmith 2001), proposing four major principles: 1) individuals and interactions over processes and tools; 2) working software over comprehensive documentation; 3) customer collaboration over contract negotiation; and 4) responding to change over following a plan. The modern definition of agile software development evolved in the mid-1990s as a reaction against the heavyweight methods, which are characterised by a heavily regulated, regimented, micro-managed use of the waterfall model of development. Methods and processes originating from the waterfall model's use are seen as bureaucratic, slow, demanding and inconsistent with the ways that software developers actually perform effective work (Fitzgerald 2000). Agile methods are sometimes characterised as being at the opposite end of the spectrum from plan-driven or disciplined methods. However, this distinction is quite misleading since it implies that agile methods are unplanned or undisciplined. A more accurate distinction is that methods exist on a continuum form adaptive to predictive and that agile methods lie on the adaptive site (Boehm and Turner 2004).

Extensive efforts have been made to outline best practices in agile methods (e.g. Lindvall, Basili et al. 2002). Some argue that agile development should be test-driven, while others claim that it is all about feedback and change (Williams and Cockburn 2003). Yet, it should be mentioned that most existing literature is dominantly written by practitioners, and occupied with "abstract principles" (Abrahamsson, Warsta et al. 2003) leading to a lack of theorisation and conceptualisation effort on agile methods (Conboy and Fitzgerald 2004). Recently, attention has been paid to organisational culture which embraces agility, and that it is the people, not the methods, which generate agility (Adolph 2006).

*Agility* is defined as "the quality of being quick-moving and nimble" (Lyytinen and Rose 2006), meanwhile it requires discipline and skill. In software development therefore, agility can be defined as "the ability of developers to sense and respond to new technical and business opportunities in order to stay innovative and competitive in a turbulent business environment" (ibid) or as One-Ki, Probir et al. (2006) argue, ISD agility represents the ability of systems development practices to rapidly adapt to the changing business requirements and environment. Unlike traditional development practices, agile approaches deal with unpredictability by relying on people and their creativity and competence rather than on processes (Highsmith and Cockburn 2001) and are adaptable to project specific circumstances, such as the experience of the team, customers' demands, etc. (Bajec, Krisper et al. 2004). Implied in agile methods therefore is the idea of improvisation (Bansler and Havn 2004**).** Agile practices view people differently than traditional practices. "While rigorous practices are designed to standardise people to the organisation, agile practices are designed to capitalise on each individual and each team's unique strengths" (Cockburn and Highsmith 2001).

Agile methods are also characterised by short iterative cycles of development-driven, collaborative decision-making, extensive communication, incorporation of rapid feedback, parallel development and release orientation, features which show their ability to respond to change and create innovation (Highsmith 2003). The deliverable of each development cycle is working code that can be used by the customer. In agile development, a project is broken down into sub-projects, each of which usually involves planning, development, integration, testing and delivery (ibid). Agile methodologies are considered to discourage documentation beyond the code, a fact that results in the transformation of explicit product knowledge into tacit knowledge. Rotation of team membership also ensures that this knowledge is not monopolised by a few individuals. Developers work in small teams with customers being active team members (Nerur, Mahapatra et al. 2005). Therefore, agile development favours a leadership-and-collaboration style of management where the project manager's role is that of a facilitator (Highsmith 2003). Furthermore, agile teams are characterised by self-organisation and intense collaboration within and across organisational boundaries. Self-organisation does not denote leaderless teams; rather these are teams which can organise again and again to meet challenges as they arise (Cockburn and Highsmith 2001). Agility therefore, requires that such teams should have a common focus, mutual trust and respect, a collaborative but speedy decision-making process and the ability to deal with ambiguity (ibid).

The concept of agile software development has become more or less synonymous with short-cycle time development (Baskerville and Pries-Heje 2004) and amethodical development (Truex, Baskerville et al. 2000) where the system evolves continuously and the development never reaches completion but continues as the system grows, evolves and changes. Truex, Baskerville et al. (2000) suggest that the amethodical view appreciates innovation and organisational change that leads to adaptation, experimentation, as well as to accidents and opportunism. They argue that organisational contexts should be understood as emergent, so requiring an ISD approach that is adaptable and flexible. Consistent to these concepts, Baskerville and Pries-Heje (2004) package agile methods as "short cycle time systems development" practices focusing on completion speed, release orientated parallel prototyping, architecture, negotiable quality and an ideal workforce. This package is consistent with amethodical development concepts that support emergent systems and thereby organisational emergence (ibid).

***Open source software development***

Open source software development (OSSD) is also often characterised as amethodical, evolving endlessly through a series of rapid releases, each developed by separate teams using disparate methods (Jorgensen 2001). The OSSD approach is argued to represent a significant alternative to modern software engineering techniques for developing large-scale software (Scacchi 2002). One of the most important features of OSSD is the formation and enactment of complex software development processes performed by loosely coupled coordinated developers and contributors that may be globally dispersed (Scacchi, Feller et al. 2006)**.**

OSS is by definition software which is developed collaboratively and users have access to the source code (Madey, Freeh et al. 2002). By working together, a community of both users and developers can improve the functionality and quality of the software (Goldman and Gabriel 2005). OSSD projects are "self-organised, employ extremely rapid code evolution, massive peer code review, and rapid releases of prototype code" (Madey, Freeh et al. 2002). OSSD is considered to be a prototypical example of a decentralised self-organising process in that there is no central control or planning and it challenges traditional economic assumptions and the principles of conventional software engineering and project management. Furthermore, it is argued that engaging in an OSSD project requires understanding of the community, its culture and customs, its tools, and its way of working (Feller and Fitzgerald 2000), as community building, alliance formation and participatory contribution are essential activities in the OSSD projects to persist without central corporate authority (Scacchi, Feller et al. 2006).

OSSD projects are argued to enact Internet time development practices, which focus on iterative development, incremental releases that are driven by feedback from users as a way to determine requirements and prioritise incremental functionality, as well as on daily system builds (Scacchi 2002). Peer reviews of the source code are developed and usually major releases undergo more review compared to the daily build releases. OSSD indeed encourages software reuse and resource sharing as well as ongoing evolution of tools and applications through re-invention as a basis for continuous improvement (Scacchi 2004a). It is claimed that such projects are oriented to community and agility rather than relying on formal project management and traditional software engineering techniques that may increase bureaucracy (ibid). Indeed, individual contributors share equal power that they gain by their successful sustained contributions over time. Interestingly, One-Ki, Probir et al. (2006) suggest that agile practices are closely aligned to OSSD practices and evoke the idea of coordinated development processes through the rational division of tasks that are voluntarily selected and developed by individuals or groups according to their interests. However, Scacchi (2004a) argues that agile methods stand somewhere in the middle ground between traditional software engineering and OSSD practices, with OSSD representing a new paradigm of global systems development.

The following section discusses the perceived "gap" in the GSD literature. Already proposed methods (e.g. agile and open source practices) are being examined but it is, however, evident that they fail to address the distributed nature of global collaborative development work. The need for new collaborative practices and guidelines for successful GSD is therefore being discussed.

### 2.2.3 Silver bullet to GSD

GSD is considered to be the new paradigm in developing large-scale systems (Damian and Moitra 2006). However, there are still challenges involved that need to be addressed for having successful GSD (Herbsleb and Mockus 2003). The globalisation increases the complexity and uncertainty of the collaborative development effort, which can in turn negatively influence project outcomes (Lee, Delone et al. 2006). Whereas the literature stresses for practices and processes that are flexible and adaptable to the increasingly volatile requirements of the business environment (Highsmith and Cockburn 2001), Lee, Delone et al. (2006) argue that successful GSD requires not only flexibility/agility but also rigor/discipline in order to cope with complex challenges and requirements of global projects. Similarly, there is an ongoing debate which rejects the idea of agile practices as a "silver bullet" to the challenges of GSD (Parnas 2006). Although agile practices can work perfectly well in small, self-organised co-located teams (Boehm and Turner 2004), there is an urgent need for scaling agility to incorporate distributed and collaborative systems development situations (Zheng, Venters et al. 2007).

A number of studies stress the inadequacy of agile practices to be used in the global development context as they are now (Agerfalk and Fitzgerald 2006; Ramesh, Cao et al. 2006; Sauer 2006; Smits 2007; Conboy 2009). As they argue, many of the key concepts within agile development such as pair programming, face-to-face interaction and onsite customers, etc. are difficult to apply (ibid). Authors suggest that agile methods need to be adjusted and modified by adding more planning and in order to embrace more rigour in software development (Lee, Delone et al. 2006; Simons 2006; Smits 2007). In co-located teams these might hinder flexibility, but without them the GSD environment becomes chaotic and inefficient. Detailed comprehensive documentation as well as codifying knowledge is also crucial in global contexts as communication is problematic and tacit knowledge is difficult to share, something that is argued that agile methods do not embrace (ibid). Research on global outsourcing and offshoring has made some effort to provide lessons that may enhance agile methods usage in global contexts (Kussmaul, Jack et al. 2004; Fowler 2006; Sauer 2006; Sutherland, Viktorov et al. 2007). Layman, Williams et al. (2006) have also attempted to enhance extreme programming (XP) by identifying factors for communication that need to be addressed in order to enable the creation of globally distributed XP teams; Flor (2006) similarly attempted to enhance pair programming to yield the same benefits in a global environment, while, Ramesh, Cao et al (2006) suggested a balancing act and proposed practices to achieve distribution and agility. Yet, those studies' findings are argued to be a first step towards a more formal investigation into using agile methodologies in GSD. Therefore, there is still a long way to go before agile practices are scaled enough to be considered as the "silver bullet" to GSD.

The OSSD methodology, on the other hand, is also open to debates for its usage in the general GSD context. Those debates mostly relate to the inflexibility of the OSSD context to promote collaborative relationships (Jensen and Scacchi 2005). Developers are expected to work in isolation without coordinating with other community members. They are supposed to offer up their work for consideration and inclusion after it is finished. It is indeed claimed that "reducing the need for collaboration is a common practice in the OSSD community that is believed to give rise to both positive and negative effects" (ibid). However, literature suggests that collaboration and collaborative practices are crucial for GSD (Yalaho 2006). It is, therefore, argued that since the OSSD context tries to reduce collaboration, then the systems development practices employed in this context perhaps promote individualistic behaviour rather than promoting collaborative practice (ibid).

GSD is argued to be "a discipline that has grown considerably richer through practice, influencing research and established practices themselves" (Damian and Moitra 2006). However, the practices and methods employed are far from being mature and fully understood (Herbsleb and Moitra 2001). With the globalisation of systems development, the limitations of traditional systems development practices become even more obvious (Hanseth and Monteiro 1998). Furthermore, there is a fundamental shift from the development of traditional IS to the development of global information infrastructures, such as the Grid (EuropeanCommission 2006). Grid infrastructures should be seen and treated as large-scale and open as they demand collaborative development in a global/distributed environment (Nentwich 2008), an environment which is characterised by high uncertainty and complexity and a continuous stream of improvisation, bricolage, drifting, mutual negotiation, regularity, progress and cycles of interactions (Nandhakumar and Avison 1999). Development often requires ad-hoc problem solving skills and creativity, skills which cannot easily be pre-planned (Ciborra 2002). GSD for large-scale systems demands new, different systems development practices that value and focus on human and social-related factors, as the nature of the problem and the environment are now different. Long-cherished computer science principles and early systems development, therefore, need to be re-examined in the light of the new requirements.

Despite the popularity of the topic, the art and science of GSD is still evolving (Agerfalk, Fitzgerald et al. 2009). Smite, Wohlin et al. (2010) claim that while a number of empirical studies and reports exist on the topic, their results are still controversial and discussed in general terms, requiring practitioners to put considerable effort in order to reflect on their practice in light of these findings. There is still no ‘standard’ or recipe for successful GSD performance and therefore it is argued that there is an urgent need to evaluate the field of GSD from an empirical perspective and provide guidance for future progress (ibid). Furthermore, various authors argue that there is a limited understanding of agility in distributed system development settings with literature highlighting that the concept of agility may be paradoxical for a distributed setting, e.g. (Simons 2006; Kettunen and Laanti 2007). This study attempts to bring some clarity on the topic, revealing that multiple dimensions of agility exist that are applicable to GSD settings. It is in these “gaps” that this study contributes by describing new “forms” or guidelines for global distributed systems development practice based on lessons learned from the PP context.

The section below introduces Part 2 of the literature review, focusing on collaborative practice and communities. Part 2 particularly introduces concepts such as CoP, colonial systems and collective mind, which can enhance the activity theoretical framework employed in this research for a better analysis of the empirical data.

## 2.3 Part 2: Global virtual communities: Collaborative practice, shared identity and learning

Collaboration and collaborative practices play a vital role in GSD. In this study collaboration is defined as a configuration of people, globally distributed, planning, deciding, acting and working together with an intended aim and purpose. It is considered as a beneficial and well-defined relationship between various actors with a commitment to a set of common goals, a jointly developed shared responsibility, and mutual authority and accountability (Mattessich, Murray-Close et al. (2001). The final work product or outcome of such collaboration reflects all participants' contributions (John-Steiner, Weber et al. 1998).

It is indeed argued that without collaboration the inherent challenges of the global development process become even more difficult to handle (Paasivaara and Lassenius 2003). It is therefore central to this research to conceptualise the nature of collaborative practice in order to achieve a better and more thorough analysis of the research findings. A number of authors have proposed different conceptualisations of collaborative practice, with key concepts being Wenger's CoP (Wenger 1998), Porra's colonial systems (Porra 1999) and Weick's collective mind (Weick and Roberts 1993). All these concepts stress the importance of sharing a set of values, norms and goals, as well as that these communities collectively develop unique social and cognitive repertoires that guide their own interpretation and practice of the world.

This section of the literature review firstly introduces a definition of practice, and collaborative practice in particular and then moves on to identify the most important characteristics of the aforementioned communities. Community for this study is defined and perceived as a learning community with a collaborative configuration of people who share an intention within their work practice and who share knowledge similar to a CoP (Wenger 1998) or a colony (Porra 1999) in order to achieve the aims of such intention. The contribution of this study however, is the identification of a new form of community, which draws upon features of CoP, colonial systems and the collective mind but which also exhibits distinctive characteristics. This is further discussed in the discussion chapter of the thesis.

***Practice and collaborative practice***

Practice is defined as a socially recognised form of activity which is being performed on the basis of what members learn from others. Practices are forms of action which suggest that when people engage in practice, they posses competence or power (Schatzki 2001). A number of theorists have defined practice as the skills, or tacit knowledge and presuppositions that underpin activities (Turner 1994). It is however argued that practice should be treated as involving both thought and action and should be seen as being enacted by people of a collective (ibid). Swidler (2001) has suggested that practice encodes patterns of action (a schema) that people not only read but also enact. This schema is what enables membership in a community and allows participants to identity with the specific group (ibid).

Diverse interpretations of practice exist with the most common definitions emphasising practice as a routine and habitual behaviour. Current conceptions of practice emphasise the often taken for granted actions of people in their work; "a routinised way in which bodies are moved, objects are handled, subjects are treated, things are described and the world is understood" (Reckwitz 2002). Thévenot (2001) indeed stresses that for a number of years sociologists have viewed practices as habits, routines, customs and traditions. Likewise, Swidler (2001) emphasises practice as habits rather than consciously chosen actions at the individual level, while at the organisational level, suggests practice to be seen as an organisational routine. Organisational routines have been defined by Feldman and Rafaeli (2002) as "recurring patterns of behaviour of multiple organisational members involved in performing organisational tasks". As Barnes (2001) argues, something is routine when it proceeds automatically and does not involve calculative intervention. Through organisational routines, connections between members are being established, leading to the development of shared understanding about what actions to be taken in a specific routine (ibid). This shared understanding, therefore, helps organisations to maintain a pattern of behaviour that coordinates the actions of individuals and adapts to changes to the environment (Feldman and Pentland 2003).

A renewed understanding of practice has been proposed by Knorr-Cetina (2001), with a specific focus on objectual or epistemic practice. Emphasising the unfolding nature of knowledge fields, Knorr-Cetina (2001), highlights that the concept of practice has to be understood as being dynamic, ad-hoc, creative and constructive. The current use of the 'practice' term emphasising the habitual and rule-governed features of practice as customary or routinised ways of behaving is argued to provide insufficient explanation of practice in today's knowledge world where "experts" deal with epistemic objects which are open, question-generating and complex and they also have to continually learn and adapt their practice to new knowledge and experience. This therefore makes practice dynamic. Knorr-Cetina (2001) uses the notion of epistemic practice to shift the attention away from mental objects and toward the reordered conditions and dynamic chains of actions in collective life.

Various authors have tried to grasp the nature of shared/collaborative practice, with some arguing that shared practice is constituted of a number of separated individual habits and therefore it can be nothing more than habitual individual behaviour. Yet others claim that shared practice is more than a summation of practices (habits) at the individual level; rather it is a collective accomplishment of competent members. Both statements though suggest shared practice as a routine activity (Schatzki 2001). While shared practices, "are accomplishments achieved by, and routinely to be expected of, members acting together, they nonetheless have to be generated on every occasion by agents concerned all the time to retain coordination and alignment with each other in order to bring them about" (Barnes 2001). Therefore, while perhaps being routine at the collective level, it is argued that it is more than routine at the individual level (ibid). Such shared practice is argued to be sustained through continuous modification and adaptation of individual responses as people interact with others and can be gained by prolonged social interaction with members of the culture that embeds the practice (Collins 2001).

A kind of collaborative practice is software development practice. Systems development practice for this study refers to the "sum total of activities involved in developing, implementing and promoting the use of software" (Floyd 2002). The term practice is used here to relate to the different well-established ways that different development companies have cultivated to cope with customer requirements, the business environment, market section and software products (Hansson, Dittrich et al. 2006). Of course ad-hoc behaviour and improvisation is necessary to handle exceptions and to maintain the "normal", but it is only perceivable by its deviation from both the formalised rules and the established practices (ibid). Software practice is not content with observing but it is concerned with bringing about change. It is itself constructive, in that it is oriented towards developing, introducing and enhancing computer artefacts. Software practice has therefore an innovative and inventive character and has to be built on basic notions of dynamics and process in understanding design, use and implementation (Bertelsen 2002). It should come as no surprise that the practice of producing software is shaped by a complex mix of the nature of the work product, the tools and techniques used, the formal and informal organisational structure and management practices, the organisational politics and power games as well as by the formal procedures and the more-or-less hidden work practices of developers (Nørbjerg and Kraft 2002).

Although various practice definitions exist, it is generally agreed that whether individual or organisational, the study of practice indicates the study of behaviour. This study considers practice and collaborative practice as epistemic, as an emergent property linked to improvisation (Weick 1998), bricolage (Lanzara 1999) and dynamic competences which unfold as large-scale projects evolve.

In recent social research there has been a renewed theoretical interest in the concept of collaborative practice (Schatzki, Knorr-Cetina et al. 2001). Concepts that emphasise the social structures that enable collective acting or knowing and are therefore trying to explain collaborative practice and its connection with issues such as identity, learning and participation have thus emerged. Most of these concepts converge on the point that actors engaging in collective practice have some level of a shared system of meaning on which meaningful collective actions can be performed and understood. Some of these concepts are now explored.

***Communities of practice***

Literature suggests that collaborative practice and learning is a social participatory activity in which CoP spontaneously emerge as mediums within which knowledge is created and shared (Lave and Wenger 1991). Collaborative practice is performed and new learning is therefore achieved though the active relationships that individuals have with others within the context of a community (Brown and Duguid 2001). Such CoP exist within organisations, alongside formal organisational structures, and their existence is considered to be a desirable aspiration for global organisations (ibid), since they are considered to be a vital social structure for generating learning and knowledge (Wenger, McDermott et al. 2002).

According to Wenger (1998), CoP are a special type of community, where members are in a continuous state of mutual engagement in common practice (e.g. sharing know-how, or developing a system). Practice is described here as collaborative because it is defined as a common way of acting, acknowledged by the community as the correct way of doing things (ibid). CoP are defined as "the context within which learning and innovation occurs and is shared" (Venters and Wood 2007). Indeed, constant negotiation of meaning, participation and reification is an evolving process within such communities and that is why CoP are perceived as highly unstable social configurations influenced by the context within which they reside and the power relations within that community. A community cannot achieve complete stability since change is an ongoing part of practice and cannot be forced.

A CoP therefore evolves through shared practice and interaction among its members, so events and perturbations can either stabilise or destabilise such a community. Unique social and cognitive repertories that guide participants' own interpretation and practice of the world are developed collectively within such CoP (Wenger 1998) and that is why identity and learning are tightened together in particular practices which give meaning to the community (Lave and Wenger 1991). The concept of shared identity is central to CoP as it is defined as the individual's connection to the community, thus the bond that creates a sense of belonging and oneness (Van Dick, Wagner et al. 2004). It is argued that such a shared identity is produced as a lived experience of participation and reification in a CoP, shaped by engaging in practice but with unique experiences and by belonging to a community but with a unique identity (Wenger 1998).

Lave and Wenger (1991) have identified the importance of legitimising 'peripheral participation' of newcomers in a CoP by gradually developing their skills and through the involvement in collaborative practices, so establishing their shared identity. Knowing and learning are relations among people engaged in everyday activities through experiences, informal talks and stories and therefore new members need to be gradually introduced to the culture of the community (ibid). Orr (1990) has demonstrated how collective expertise could be developed through storytelling which is a way to humanise situations that occur in an organisational context. Stories give meanings to shared experiences (Brown and Cook 1999) representing symbolic means by analytic or metaphoric interpretations. In this regard, personalised directories that are developed communally over time in interactions among individuals in the group and that exist more or less complete in the head of each group member who has been completely socialised in the group may offer opportunities to remote counterparts to develop, manage and coordinate the collective expertise of the entire team through encoding, storing and retrieving activities. This way, members can reflect on their own practice by legitimising a particular way of learning that carries its own peculiarities, which may also turn to critical self-reflections. This concept of situated learning therefore reinforces collaborative practice and is vital to understanding CoP.

The emergence of new ICT and the Internet has provoked the creation of various virtual communities where communication is performed for the most part, or even solely by means of ICT (Bourhis and Dube 2010). This has resulted in the emergence of various types of electronic communities (Teigland 2000) where individuals can share their organisational knowledge; some of these can be referred to as ‘virtual CoP’ (McLure Wasko and Faraj 2000; McLure Wasko and Faraj 2005). Engendering CoP is seen as an innovative way to manage knowledge and sustain innovation (Swan, Scarbrough et al. 2002). The existence of such ‘virtual CoP’ is an important aspiration of global organisations since they are considered to be a vital social structure for generating learning and creating a shared identity (Wenger, McDermott et al. 2002). Global organisations and in particular GSD projects lack direct control and coordination and indeed the literature suggests that such traditional controls should be replaced by softer mechanisms with one vital mechanism being the formation of a shared identity (Wiesenfield, Raghuram et al. 2001) a key inherent assumption of CoP (Wenger 1998). Such a shared identity is argued to correlate with work effort, willingness to perform extra-role behaviours and task performance (Dutton, Dukerich et al. 1994), as well as promote a sense of “togetherness” in virtual groups, despite the lack of physical proximity and shared context (Yan 2008). Therefore, through its impact on employees’ motivations, a shared identity facilitates coordination and control without the need for costly (possibly ineffective) systems of supervision and monitoring (Wiesenfeld, Raghuram et al. 1998), and mitigates the challenges of distributed collaborative work by creating more trustful environments (Mannix, Griffith et al. 2002). Organisations nowadays therefore show an increased interest because of the possibility to take this old concept to address today's global environment challenges (Rios, Aguilera et al. 2009).

Nevertheless, working in a distributed and virtual environment places strains on the way a CoP works as they not only have to cope with geographical distribution, but with time, cultural and language differences as well as effective knowledge sharing (Fang and Chiu 2010) as not all knowledge can easily be captured, codified and stored (Kimble and Hildreth 2005). At present there are still scholars who are opposed to this idea and the term 'virtual CoP' is not widely accepted (Kimble and Hildreth 2004). Although some authors, for example (Hildreth, Kimble et al. 2000), describe CoP in a geographically distributed sense, they eliminate the notion of virtuality. For many authors therefore the question "Can CoP be virtual?" is vital and it has become, in a sense, a peculiar 'philosophers' stone'. While Wellman and Gulia (1999) were among the first to conclude that 'virtual CoP' could exist, at present there are still a number of scholars against this idea and the term 'virtual CoP' is not widely accepted (Kimble and Hildreth 2004; Engestrom 2009); some authors refer to it as 'On-line CoP' (Cothrel and Williams 1999), 'Computer-mediated CoP' (Etzioni and Etzioni 1999), 'Electronic CoP' (McLure Wasko and Faraj 2000), 'Networks of practice (NoP)' (Brown and Duguid 2001) and 'Distributed CoP' (Wenger, McDermott et al. 2002) even though all of them are describing a similar concept.

This dispute is formed around the first usage of the term CoP by Lave and Wenger (1991) where situated learning in co-located settings and face-to-face interaction among participants is seen as crucial in constructing a CoP. A CoP produces shared artefacts such as tools, stories and procedures that reify something of its practice. These have knowledge embedded in them and the distribution of such tacit knowledge itself requires co-location (Kimble and Hildreth 2005) and therefore geographic distribution challenges inherent basic assumptions of traditional CoP. This further raises the question as to the extent that a CoP can become distributed and work in a virtual mode and not merely be a group with links to individuals. A number of scholars, for example (Hildreth, Kimble et al. 1998), argue that CoP can function in a distributed environment, however, co-located cores are still needed. It is interestingly argued that whether a CoP can be virtual or not, is determined by the type of knowledge to be shared (Hildreth, Kimble et al. 2000). If for example the community has to be co-located because they share the same resources or documents, then virtualisation is possible, but if the nature of learning is situated due to essential face-to-face interactions then virtualisation can only be partially achieved due to the disembodiment in virtual environments (Dreyfus 2001). Nevertheless, with the rapid development of modern technologies, the former statement is arguable and it is still to be seen whether global groups can indeed be perceived as ‘virtual CoP’, or whether we are seeing the emergence of something different.

***Colonial systems***

Porra’s (1999) work on colonial systems facilitates an understanding of collaborative practice and membership in communities and how these are influenced by their culture and history**.** Accounting for the need to conceptualise collaboration and by proposing animal colonies as a metaphor, Porra (2010) attempts to understand and explain the characteristics and the processes through which groups participate in collaborative practice within typical and more recently within global/virtual organisational boundaries. She suggests, that collectives formed by people may evolve comparable to other species and that is why she defines such collectives as colonies which are perceived to be communities where a voluntary collection of individuals who share a history and an environment participate in collaborative practice and cooperate in their attempt to maintain their colony (Porra 1999).

A colony is, therefore, argued to be the product of a long history and this history influences the current colony and the way it works. The colony's members are volunteers who share a unique perspective on the colony’s past and local conditions and they collaborate to maintain stability or to carry out radical changes during environmental shifts (ibid). While individuals within the colony may exhibit individual characteristics, the essential nature of the colony is still a collective phenomenon and their collective evolutionary history plays a vital role in the survival of such a colony (Porra 1996). Colonial systems are therefore defined as “systems that facilitate the formation of collections of interdependent members and evolution of characteristics that maintain collective stasis and can create or respond to change through collective awareness (ibid).

Colonies are argued to be unique and have unique characteristics because of their unique history, which is widely shared between its members. Whereas, some colonies may portray collaborative behaviours, others can be more individualistic (Porra 1999). This culture of collaboration or individualism is passed on from generation to generation as customs, norms, values, stories and behaviour (Porra 2010). It is therefore designed into the community as rules of conduct. The essence of inheritance, which is similar to CoP (Wenger 1998), is vital to such colonies, since practice, knowledge, values, culture, norms etc. are essential components the next generation inherits (Porra and Parks 2006). Porra (1999) further argues that unlike formal organisations that are based on career and monetary contracts, colonies are formations that are based on seemingly spontaneous acts for achieving a shared goal or ideal, they have the ability to evolve and are founded on bindings such as long-lasting common interests and shared identity. Such colonies may appear leaderless without clear authorities for goal settings, power or control (Porra 1996). Therefore, there is a lack of hierarchy and individual-centred leadership. Nevertheless, no other outside entity can impose its authority over a colony without its consent.

It is indeed suggested that colonies share similar characteristics with CoP (Wenger 1998). Colonial boundaries resemble boundaries of CoP since they are unclear and the colony has no maximum size or rate for growth. Furthermore, the more time members spend together, the more they absorb the identity of the community and the more fully they are able to participate (Porra 2010). However, in contrast to CoP, colonial boundaries are argued to help bridge virtual community problems, since the more time the community spends interacting in e.g. an online forum, the more its humaneness reflects the features of this forum (Porra and Parks 2006).

The colonial systems perspective is argued to allow for a better understanding of organisations as environments in which human colonies evolve as independent entities throughout and within organisations, with their own systemic quantities and mechanisms and share histories, identities and destinies of their own (Porra 2010). Creating shared identities is indeed a vital characteristic of colonies and this further adds to the understanding of complexities involved in sustaining and managing virtual communities (Porra and Parks 2006). It is therefore suggested that the concept of human colonies provides a novel perspective on human collectives in real and virtual settings (ibid). Nevertheless, while the colony notion describes virtual communities well, it is argued that it best describes virtual communities that their members also meet face to face (ibid). Although the growth of the Internet today allows sharing colony-like experience, such hybrid communities tend to have stronger identities than pure online communities and they are able to bond better and share values and knowledge more effectively (Etzioni and Etzioni 1999). Thus hybrid communities, which represent a large number of virtual communities today (Ward 1999) are thus argued to resemble more colonial virtual communities.

***Collective mind***

Weick and Roberts's (1993) collective mind forms another interesting conceptualisation of collaborative practice which is enacted when individuals engage in a social activity and construct mutually shared fields to respond to a certain event. For Weick and Roberts (1993), the word "collective" refers to individuals who act as if they are a group, interrelate their actions and focus on the way this interrelating is done. Although a collection of individuals contributes to the collective mind, still this collective mind is different from an individual mind for a number of reasons. Firstly, it involves the pattern of interrelated activities between many people and secondly, these interrelations are being continually constructed by individuals through the ongoing activities of contributing (e.g. loose-coupling, diversity), representing (e.g. mutual respect, coordination) and subordinating (e.g. trust) (ibid).

It is argued that individuals need to collectively make sense of a situation and gain collective competence in order to be able to act together and participate in collaborative practice as a group rather than as individuals (Weick 1993). It is however claimed that shared experiences play a vital role in collective sense-making and therefore in the creation of the collective mind (Nielsen 2003) something which makes individuals interdependent on each other, since they always have to take into consideration what others think and how others will act.

The collective mind is argued to bring forward important aspects of collaborative practice, such as the need for shared sense-making, the importance of high interaction with newcomers in order to interrelate to such collective mind as well as the creation of shared experiences and it is itself presented as a complex mixture of people's past experiences, know-how and current practice with people's capability of acting heedfully (thus their capability of acting consistently, attentively, vigilantly, conscientiously and pertinaciously) (Weick and Roberts 1993). The collective mind is suggested to be ultimately "more capable of intelligent action and comprehension of unexpected events the more heedfully the interrelating is done" (ibid). Nevertheless, such collective mind needs to continually evolve, suggesting the group of people involved to redo the patterns of interrelation as they grow older, in order to "renew" the mind itself and therefore regain their capability of collectively dealing with unexpected events.

The usefulness and feasibility of creating a collective mind in global organisations is still a pertinent question to be answered. A number of scholars, however, have attempted to address such a question, with most suggesting that it is feasible for a collective mind to be developed in distributed and hybrid organisational settings (D'Eredita, Misiolek et al. 2005; D'Eredita and Chau 2005) and that it can indeed benefit such settings because the development of a collective mind represents a high-order learning in virtual team settings (Yoo and Kanawattanachai 2001).

The final section of this literature review introduces the Grid technology as an e-information infrastructure of a distinctive nature. The focus of this study is the development of the Grid. However, little research has been done on Grid development. This section therefore defines what the Grid is for this study.

### 2.4 Part 3: Introducing the Grid

Advances in distributed computing, high quality networks and powerful and cost-effective commodity-based computing have given rise to the Grid computing paradigm (Baker, Buyya et al. 2002). The popularity of the Grid has been rapidly growing, as it promises to give new impetus to the IT market and improve growth and competitiveness, by enabling knowledge and computing resources to be delivered to, and used by, citizens and organisations as traditional utilities (like the electricity Grid) and thus change significantly the life of individuals, organisations and society, as the Internet has done in the past decade (Chetty and Buyya 2002; Smarr 2004; Carr 2008).

Grid technology indeed promises the ability to distribute and share computing and data resources on tap in a seamless and dynamic way as well as to facilitate transparent collaboration within groups and across borders (Foster 2003). The Grid is often seen as "an emerging network-based computing infrastructure providing security, resource access, information and other services that enable the controlled and coordinated sharing of resources among VOs formed dynamically by individuals and institutions with common interests" (Foster, Kesselman et al. 2001). A Grid's VO define what is shared, who is allowed to share, the conditions under which such sharing occurs and present different capabilities and resources to users as defined by their membership to a VO (Foster and Kesselman 2004). These are then encoded as rules within a Grid's technical infrastructure (ibid). Grids are centred round a set of standards for the control of distributed resources that are realised as Grid middleware software. Just as Internet standards enable the sharing and integration of information on the Web, so Grid protocols aim to allow the integration of not just information, but sensors, applications, data-storage, computer processors and most other IT resources (Wladawsky-Berger 2004).

Grids offer a practical solution to the problems of storing, distributing and processing the large amounts of data that are, or will be, produced by industry and scientific communities (Foster, Kesselman et al. 2001; Hey and Trefethen 2002). They are therefore central to the undertaking of advanced science such as PP (Venters and Cornford 2006) and biomedicine (Amendolia, Estrella et al. 2004) and in industrial areas such as financial services and engineering (Wouters and Schroder 2003). The Grid was chosen as an appropriate technology for e-Science, as it represents the new paradigm for the infrastructure that will enable scientists to share and analyse results, tackle more complex problems, ask bigger questions and foster collaborations between geographically dispersed institutes (De Roure, Jennings et al. 2003).

This study does not consider Grid development and adoption as merely technical components requiring implementation, but rather as integrated socio-technical achievements similar to information infrastructures (Hanseth and Monteiro 1998). Information infrastructures are the mixture of the complexity of the IS used in an organisation and the organisation's practices and routines within which they are situated (Cordella 2006). While Grid developers (e.g. particle physicists) may aspire to create a machine, they are rather building a socio-technical infrastructure (Hanseth and Monteiro 1998) established through complex socio-technical processes and shaped by events, circumstances and unpredictable courses of action during development and use (Broadbent and Weill 1999; Cordella 2006).

People are a vital component of this e-information infrastructure (OSIgroup 2007) and such people's expectations, prejudices and interpretations can shape the way the Grid is being developed as well as how the Grid will be used (ibid). Indeed such information infrastructures are the output of the recursive dynamic interaction between technologies and people, and are designed as an extension and improvement of an already existing installed base of infrastructure (Hanseth 1996). The installed base affects the possible paths of development of the new elements, and therefore becomes self-reinforcing (Star and Ruhleder 1996). Like an information infrastructure, the Grid is being built upon the existing infrastructure of the Internet.

Yet Grids are perhaps more than an information infrastructure which becomes transparent in use and only become visible upon failure (Ciborra 2004). Grids have a tendency to have a strong architectural form embedded within, and shaping, their VOs (Berman, Geoffrey et al. 2003). This reflects their potential use and does not simply extend or improve the existing infrastructure, but enables a re-scaling of such infrastructure in order that new forms of practice may be enabled (Ciborra 2004). It is unsurprising therefore that the Grid is claimed to provide more functionality than the Internet on which it rests representing not just a next generation Internet (Jirotka, Procter et al. 2005), rather a fundamental step towards the realisation of a common *service-oriented infrastructure* for on-demand distributed supercomputing, high-throughput computing, data-intensive computing and collaborative computing, based on open standards and open software (Foster, Kesselman et al. 2003).

The concept of information infrastructures is a stepping point for this research to look at the Grid. Taking an information infrastructure perspective (Ciborra and

Associates 2000; Hanseth 2000) enables a better appreciation of the complex nature of the Grid as large-scale and open, of its development as collaborative, distributed and dynamic as well as of the intertwined relationship between the Grid and the people involved in its life cycle. The term information infrastructure denotes that indeed such Grids are more than individual components and “pure technology” and their successful development and deployment requires more than a combination of traditional approaches for the development of IS (Hanseth and Monteiro 1998) as the development of a Grid is rarely built in an entirely top-down and orderly-like way and it is inevitably related to the processes and practices of those involved as well as embedded in a complex web of socio-material relations (Edwards, Jackson et al. 2007).

However, while traditional accounts of the development of infrastructures see infrastructures as developed and deployed by centres of power exercising strong architectural control and focus on the need for alignment with organisational objectives, concentrating on the need for powerful actors to ensure interoperability, usage and access to the infrastructure (Weill and Broadbent 1998), this view does not reflect many contemporary forms of information infrastructure which span organisational settings; face customers who are far from submissive “users”; and are bound up in political agreements and relationships between many distributed organisational actors (Kyriakidou and Venters 2009). If Grids are indeed to support global collaborative working there must be a better appreciation of the social and political context of the Grid’s conception, construction and operation (Voss, Mascord et al. 2007) and of the Grid’s related socio-technical infrastructures (e.g. support structures, maintenance and training).

## 2.5 Chapter summary

This chapter presented a thorough review of the relevant literature on GSD, collaborative practice and communities and the Grid, and therefore has placed this study in context. The chapter consists of three relevant parts. Part 1 discussed distributed systems development mostly focusing on 1) the characteristics of such process and of the global virtual teams involved in such development, 2) the challenges such global projects face as well as on 3) already proposed solutions to these challenges. A number of contemporary systems development paradigms (agile development and open source development) that are relevant to this study as they reflect a new collaborative form of systems development were also discussed in terms of their suitability to the GSD environment. These discussions revealed the gap in the ISD and GSD literature calling for new systems development paradigms, guidelines and strategies for successful global systems development. It is in this "gap" this study contributes to.

Part 2 focused on conceptualising collaborative practice through concepts such as Wenger's (1998) CoP, Porra's (1999) colonial systems and Weick and Robert's (1993) collective mind. Such literature was reviewed in order to enhance this study's chosen theoretical framework in analysing the research findings. The community element in AT does not reflect well the complexities of the community being studied here and the theory has not previously been used to address the kind of collaborative GSD addressed in this study. Therefore, this study drew upon literature on communities to inform the theory. Such literature was also vital in constructing this study's framework of guidance for engendering HEAD_SDC.

Part 3 finally introduced the Grid and conceptualised it as a large-scale, open, distributed and dynamic e-information infrastructure, "demanding" collaborative development and practices, which are flexible and can adapt to its complex and evolving nature.

The next chapter presents a thorough overview of AT and its conceptual tools applied in this study.

# 3 Activity theory and global systems development

## 3.1 Introduction

Drawing upon the literature review of the previous chapter, this chapter describes the theoretical concepts employed in this study in order to explore and examine the collaborative distributed systems development of the PP Grid. This study focuses particularly on the collaborative systems development practices employed to facilitate the Grid's development. This collaborative, global, multi-disciplinary development effort is seen by physicists as a challenging learning journey where conflicts, competition, history and culture, collaboration and shared visions are seen as internal elements of the process and need to be accounted for. This study surfaces the collaborative practices, which emerge from the resolution of such internal struggles and contradictions of the process. Based on these new practices it draws valuable lessons that can inform the wider GSD context. Given the objectives of this study, therefore, it is imperative to approach the task with a theory that brings collaborative practice and learning to the forefront. A better understanding of distributed systems development can only be achieved by appreciating the collaborative context with its conflicts and contradictions; but also by appreciating the sometimes contradicting development actions which initiate learning activities that lead to the production of new knowledge and new more flexible collaborative systems development practices.

Previous research on ISD, or its use, or of the social impact of technology in the IS field, drew heavily upon sociological and social theories such as actor network (Latour 2005) and structuration theory (Orlikowski 1992), theories of social construction of technology (Pinch and Bijker 1984) and various organisational and cultural theories. Theoretical models from social psychology have also been widely used by IS researchers as theoretical foundations to explain and foresee ISD and use (Ditsa 2003). Most of these models and theories, however, ignore the social context (e.g. collaboration, history and time) in which IS are developed and used – characteristics which are central to this study. Such theoretical models largely centre on phenomena such as the individual and the technology, and it has been argued that they generally "suffer from a dichotomy between the individual and the social" (Kuutti 1992b). For example, if a social system is used as a unit of analysis, then problems in maintaining human agency emerge, while if individual actions are studied, then problems in maintaining contextuality arise (ibid). Furthermore, theories such as actor network theory "turns all actors into black boxes without identifiable internal systemic properties and contradictions, having as a result not been able to realise the dynamics that can energise serious learning efforts" (Engestrom 2001), something which precludes the application of this theory to this study given the importance of surfacing contradictions which energise learning efforts leading to the creation of new collaborative systems development practices.

In contrast to the deterministic views inherent in much of the literature on Grids, AT is here employed to examine how the Grid technology is collaboratively constructed to fulfil the objectives of a global community (Nardi 1996). AT has inspired a number of theoretical reflections on what ISD is about (Kuutti 1991; Bertelsen 2000). AT provides a theoretically founded but detailed and practicable procedure for studying systems development as a real-life collaborative work activity in context (Korpela, Mursu et al. 2002) rather than as an individualistic process of interaction devoid of such social context, a central element in the case of a "unique" collaborative social context under study here. AT arose as a response to the pressing need to enlarge the research object of systems development, e.g. to take better account of contextuality (Spasser 1999). Through the years the theory has undergone continual development and today it not only emphasises the centrality of practice as doing and activity, but also foregrounds setting and context as essential orienting concepts (ibid). As argued by a number of activity theorists (e.g. (Kuutti 1996; Korpela, Soriyan et al. 2000; Mursu, Luukkomen et al. 2007), the solution offered by AT is the unit of analysis, which is called an "activity" – the minimal meaningful context for individual actions –, and is better defined and more stable than just an arbitrarily selected context, but also more manageable than a whole social system. Because the context is included in the unit of analysis it makes the object of the research essentially collective, even if the main interest lies in individual actions (Jonassen and Rohrer-Murphy 1999).

While other social theories tend to predicate interaction upon agency, e.g. structuration theory (Barnes 2000), or upon social structure (Bohman 1999), in AT, practical activity is posited as the essential site for analysing interaction between actors and collective structures (Jarzabkowski 2003). Practical activity comprises a series of actions but it is a more historically situated and collective notion than any single action; rather a stream of actions is invested with meaning and purpose through the taken for granted and highly contextualised rationale of the activity system (ibid). AT emphasises the institutionalisation of social interaction, very much like the institutional theory of Berger and Luckmann (1966). It however differs from the social construction approach in its emphasis on the object orientedness of human activity and the central role of artefacts (Virkkunen and Kuutti 2000). It also "transcends the tension between social constructivism and determinism by regarding humans and human practices as simultaneously relating to the natural-material realm through tools and to the social-realm through the culturally and historically shaped collective activity" (Foot 2001). Furthermore, in contrast to traditions of material and social determinism, AT regards individuals and collectives as having agency (ibid). In activity theoretical perspective, and consistent with Vygotsky's view of learning, humans do not solely internalise or appropriate the cultural-historical and material resources available to them, but they also externalise or create new social and material forms of relations and tools in expansive cycles of development (Engestrom 2004). The explanation of social change and cycles of development by inner contradictions of activity systems is original to AT (ibid) portraying the theory's relevance to the objectives of this study.

As claimed by Bratteteig and Gregory (2001) AT provides a useful starting point for analysing the collaborative practices involved in ISD, since it considers "the elements of an activity system as dynamic and perceptually open to change" and emphasises "mediation and multiplicity of artefacts". AT also provides a well developed framework for analysing the complex dynamics of collaborative settings, such as the PP community, which typically involve interacting human and technical elements (Crawford and Hasan 2006). The concept of collectiveness and the notion of different actors sharing the same goals and constructing the same meanings are at the core of this theory (Leont'ev 1978), and are vital to the analysis of an "exceptional" community under study here. In addition, AT provides a different lens for analysing learning processes and outcomes for the purpose of designing systems as it focuses on the activities in which people are engaged, the nature of the tools they use in those activities, the social and contextual relationships among the collaborators involved, the goals and the intentions of those activities as well as on the objects or outcomes of those activities (Jonassen and Rohrer-Murphy 1999). Changing systemic interpretations about the nature of activity are thus associated with change in practice (Jarzabkowski 2003), something important in this study as it allows for the examination of how practice is transformed in facilitating such collaborative distributed development. In conclusion, AT's focus on accumulating factors that affect the subjective interpretations, the purpose, and sense-making of individual and group actions and operations, provides a useful paradigm for the ways in which human experience, needs, collaborative practice and creativity shape the development and effectiveness of emerging technologies (Crawford and Hasan 2006) and therefore make this theory suitable for this study.

This chapter presents a review of AT including an explication of how its core principles convincingly suggest its suitability as a framework for studying distributed systems development of large-scale systems. A brief overview of AT is presented in Section 3.2 in which the fundamental principles which underpin its development from Vygotsky (1978) through Leont'ev (1978; 1981) to Engestrom (1987) are discussed. AT's application in IS is then discussed in Section 3.3. The sections that follow present the theory's conceptual tools applied in this research. Section 3.4 therefore presents the activity triangle after which the general structure of an activity is reviewed in Section 3.5. Contradictions as conceptualising breakdowns and tensions in collaborative activity are then discussed (Section 3.6) after which a thorough discussion of how their resolution can facilitate change and expansive learning is presented (Section 3.7). Section 3.8 summarises the key concepts discussed.

## 3.2 Overview of AT

### 3.2.1 Generations of research in AT

AT originated in the former Soviet Union around the 1920s-1930s as a part of the cultural-historical school of psychology founded by Vygotsky and his colleagues Leont'ev and Luria (Adams, Edmond et al. 2003). Vygotsky is considered to be the major theorist among the social constructionists (Huitt 2003) and AT is therefore based on the constructionist beliefs (ibid). Mahoney (2004) argues that the theories expressing constructionism have five basic themes: (1) active agency, (2) order (patterning of experience by means of a tacit meaning-making process), (3) self (the self is not an “isolated island”, since people exist and grown in living webs of relationships), (4) social-symbolic relatedness (individuals cannot be understood apart from their organic embeddedness in social systems) and (5) lifespan development. AT is based on the constructionism ontology, since it focuses on the acting subject’s potential to create reality, but at the same time, AT tries to understand learning communities that are more closely related to the collaborative practice of the real world (Engestrom 2000a). Therefore, AT tries to overcome the individualism, and attempts to focus on the historical and collaborative construction of knowledge and reality by people and their artefacts within the social world (ibid).

AT has evolved through three generations of research (Engestrom, Miettinen et al. 1999). The first generation is centred around Vygotsky (1978), whose broad aim was to create a developmental psychology drawing upon Karl Marx’s ideas. His goal was to explain human learning activities within which he posited a learning subject and a learning object mediated by tools and signs (ibid). This idea of mediation was diagrammatically presented in the famous triangular model in which the conditioned direct connection between subject and object was transcended by a complex mediated act (Figure 3.1.). In any conscious human activity, there is a *subject* who pursues an object. The object is what is being worked on, e.g. it is not the objective, rather it is what is being shaped and transformed by the tool, while the tools can be material artefacts language or concepts and are inscribed with what is culturally important (ibid).

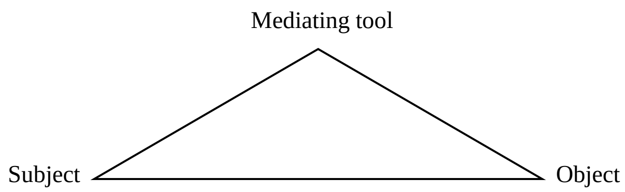


Figure 3.1 Vygotsky’s model of mediated act

The insertion of cultural tools and signs into human action was revolutionary in that the basic unit of analysis overcame the split between the individual and the untouchable societal structures (Daniels 2005). The individual could no longer be understood without their cultural means and the society could no longer make sense without the agency of individuals who use and produce these tools (ibid). This idea however, challenged existing Piagetian notions of learning as cognitive development (Pass 2004), and objects thus ceased to be just raw materials for the formation of logical operations in the subject as they were for Piaget; rather, objects became cultural entities and the object-orientedness of action became the key to understanding human practice (Rieber and Wollock 1997).

The limitation of the first generation was that the unit of analysis remained individually focused (Engestrom 1999b). Mediation by other human beings and social relations was not theoretically integrated into the triangular model of action. However, activity theorists attempted to overcome this in the second generation, which centred round Leont’ev’s research. Leont’ev and his group shifted the focus of their research away from tool mediation and towards the object, on how it was interpreted and what actions it elicited (Leont'ev 1978).

> “The main thing which distinguishes one activity from another is the difference of their objects. It is exactly the object of an activity that gives it a determined direction. According to the terminology I have proposed, the object of the activity is its true motive” (ibid).

In this generation, the crucial difference between an individual action and a collective activity was explicated and the activity concept was further developed by elucidating its constituent operations–actions–activity levels (Leont'ev 1981). However, Leont’ev never graphically expanded Vygotsky’s original model into a model of a collective activity system. For Leont’ev all activities are social, even those carried out in apparent isolation; however his work’s focus was on ‘concrete individuals’ engaged in individual activity (Hyssalo 2005). Although Leont’ev saw individual activities as necessarily social, this does not entail that individual activities are necessarily collective (ibid). Nevertheless, Leont’ev’s work does not discount the possibility of collective activity and Engestrom (1987) made a good case for reading his work as a move towards collective activity. Leont’ev’s work was expanded by Engestrom (1987), who elaborated a framework (Figure 3.2) for an activity system which helps map relationships between, for example, tools, object and rules.

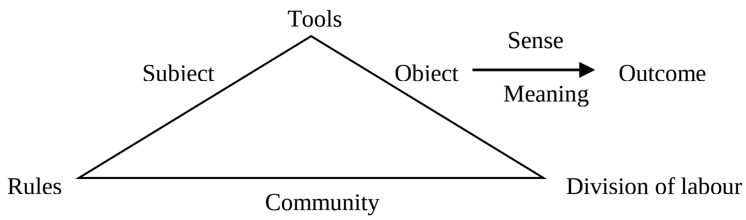


Figure 3.2 Human activity system (Engestrom 1987)

Engestrom built upon Leont'ev's work in developing the concept of activity as a unit of analysis. Engestrom, based on Vygotsky's requirements, elucidated a definition of activity as a unit of analysis that fulfils the following demands: it is representative of the complexity of the whole, it is analysable in its contextuality, it is specific to human beings by being culturally mediated and it is dynamic rather than static (Engestrom 1990). Engestrom further critiqued Leont'ev on the basis that the instrumental and communicative aspects of activity were not brought into a unified complex model. While accounting for hierarchical levels of human functioning, Leont'ev's theory does not go far enough to situate human functioning in context, illustrating how individual actions are transformed into shared collective objects through interactions with community members or indeed how division of labour impacts on individual actions in a collective activity. This is where Engestrom's (1987) conceptualisation of an activity system as the basic unit of analysis serves as a useful heuristic for situating cognition in context. While accepting Leont'ev's hierarchical levels of human functioning, Engestrom moved the theory forward by situating it more fully in context and focusing on the collective nature of all activity. Engestrom's work expanded the unit of activity to include three additional components that explain the social structure of activity: i) rules that regulate the subject's actions toward the object and relations with other participants in the activity, ii) the community of people who share an interest in and involvement with the same object and iii) the division of labour which defines what is being done by whom toward the object, including both the horizontal division of tasks and the vertical division of power and positions, access to resources, etc. (Foot 2001). The fundamental unit of analysis therefore became the entire activity system in the collective context (Guy 2003).

Without a doubt, AT has had its share of criticisms from other psychologists. These criticisms were mostly centred on the perceived differences between the first two generations of AT, e.g. between Leont'ev's tool mediation and Vygotsky's sign mediation. Leont'ev's seminal works on AT have been criticised for a supposedly rigid and restrictive emphasis on tool-mediated production of objects as the prototypical form of activity (Engestrom 1999a). It is argued that communication and mediation by signs is suppressed in his version of AT and his ideas deviate from Vygotsky's original views. It is true that while Leont'ev shared Vygotsky's views on learning through mediation by cultural-historical factors, he took a different view on mediation between the subject and the objective world. But the essential viewpoint in AT is not the trivial difference between the ideas of Vygotsky and Leont'ev, the essence is on how it offers itself as an instrument for studying how people learn and develop skills. In fact, in his discussions on mediated activities, Vygotsky (1978) stressed that "tools and signs are mutually linked and yet separate". However, a careful reading of Leont'ev's work reveals that both mediation by signs and subject-to-subject relations do play an important role in this theory (Engestrom, Miettinen et al. 1999). The principles these two authors commonly agree on far outweigh those that are perceived to separate them.

Other criticisms involve Cole (1988) who was one of the first to clearly indicate the deep-seated insensitivity of the second generation AT towards cultural diversity. More recent criticisms include Kaptelinin and Nardi (2006) who argue that more conceptual work is needed to advance the understanding of goals in activity as the subject cannot be aware of all possible goals at the same time, as well as the understanding of individual and collective planes of activity and how one influences the other; Daniels and Warmington (2007) suggesting for power relations to be better accounted for; and Roth (2007) who argues for the need to incorporate the concepts of emotion and identity to the theory. In an attempt to enhance AT and provide a new more complete unit of analysis for the study of collaborative work, Engestrom (2001) has set up an agenda for future developments encompassed under the third generation of AT.

> "The third generation of AT needs to develop conceptual tools to understand dialogue, multiple perspectives and voices and networks of interacting activity systems. In this mode of research, the basic model is expanded to include minimally two interacting activity systems" (Engestrom 2001).

This new generation of AT particularly represents the need for the development of a common object in activity that involves different, interacting activity systems. Engestrom proposes a new unit of analysis for AT – networked activity systems whose work concerns a shared object (Figure 3.3). The expansive transformation of contradictions thus involves not just internal transformation within activity systems, but also transformations in the relations between systems.

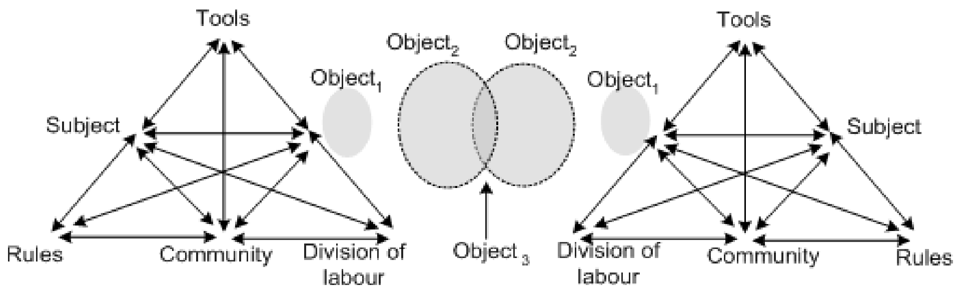


Figure 3.3 Interacting activity systems as the minimal model for the third generation of AT (Engestrom, 2001)

This idea of networks of activity within which contradictions and struggles take place in the definition of the motives and object of the activity calls for an analysis of power and control within developing activity systems (Daniels and Warmington 2007). The third generation of AT is still under development and there is an urgent need for new conceptual tools to be constructed in order to facilitate the analysis of today's environment which is global, highly distributed, open and interacting. Indeed, Engestrom (2009) concludes that the task of AT today is to create new concepts to make sense of this new global and "online" generation we live in and "to bring together the big and the small, the impossible and the possible, the future-oriented activity-level vision and the here-and-now consequential action".

A further discussion of AT's basic principles follows in Section 3.2.2.

### 3.2.2 AT's fundamental principles

AT presents a collection of basic ideas for conceptualising both individual and collective practices as developmental processes of the context in which human activities take place (Kuutti 1996). It provides appropriate conceptualisations and theorising of collaboration, a core idea of this study, through its focus on the developmental transformations and dynamics in collective activity (Bardram 1998). The core concept of the theory is that the relationship between an individual and the world is mediated by their activities, that is, their interactions with environmental objects (Engestrom 2000b). An activity entails a complete system of human practices, which are purpose-driven activities, explicit or inexplicit methods for carrying out activities, and physical and conceptual tools used as mediators when executing activities (Mwanza and Engestrom 2005). AT is deeply contextual and oriented at understanding these historically specific local practices and their objects, the mediating artefacts and the social organisation and attempts to explain qualitative changes in human practices over time (Engestrom 2003), thereby enabling an analysis of how PP practice has transformed over time to facilitate the Grid's development. Identifying changes and contradictions that exist in an activity is crucial as they serve as the means by which new knowledge about the activity being examined emerges and new tools, rules etc. are created (ibid).

The concept of different actors sharing the same goals and constructing the same meanings are at the core of this theory (Leont'ev 1978). Knowledge is not a fixed object; it is rather constructed by individuals through their own experience of the activity (Bardram 1997). When people work collaboratively in an activity in the context of a community, they bring their own perspectives to this activity and are able to negotiate and create meanings and solutions through shared understanding. AT therefore allows us to "emphasise meaning-making through active participation in socially, culturally, historically and politically situated contexts" (Kuutti 1996).

AT, is one of the few approaches that allows us to study the complexity of the development of technologies across multiple sites and scales of analysis, as well as study multiple interacting activity systems over time (Lehenkari and Hyysalo 2003). It is particularly suitable to be used in studying contextually embedded interactions, as it contains features such as the recognition of actors, mediations, historicity, constructivity, dynamics, etc. (Kuutti 1996). Interaction provides an interpretative basis from which individuals attribute meaning to their own and others actions and so are able to engage in a shared activity. The shared activity is argued to be practical, as it is conducted with an outcome in mind and the context of this practical activity is defined as an activity system (Engestrom, Engestrom et al. 2002).

In its current shape, AT can be summarised with the help of five key principles**.** The *first principle* emphasises the collective, tool-mediated and object-oriented activity system seen in its network relations to other activity systems as the prime unit of analysis. Activity systems are self-organised and reproduce themselves by generating actions and operations (Engestrom 1999b). Goal-directed actions as well as automatic operations are subordinate units of analysis and are understandable only when interpreted against the background of the entire activity system **(**ibid**).** Objects of activities are potential outcomes that motivate and direct activities, around which activities are coordinated, and in which activities are crystallised in a final form when the activities are complete (Kaptelinin and Nardi 2006). Objects can be physical things or ideals and separate one activity from another. Concrete actions can be assessed as to whether or not they help in the accomplishment of the object. But objects do not unilaterally determine activities; it is the activity in its entirety, the subject-object relationship that determines how both the subject and the object develop (ibid).

The *second principle* is the multi-voicedness of activity systems. Multi-voicedness indicates the community of the multiple points of view, traditions and interests within the activity system. As Engestrom (2001) argues, "the division of labour in an activity creates different positions for the participants, the participants carry their own diverse histories and the activity system itself carries multiple layers and strands of history engraved in its artefacts, rules and conventions". As he further stresses, this multi-voicedness is a source of both trouble and innovation and demands actions of translation and negotiation.

The *third principle* is historicity. AT incorporates strong notions of intentionality, history, culture, mediation, collaboration, practice and development over time, thus emphasising the historical and environmental context for understanding the activity's life (Kaptelinin 1996). Activities are not static things, rather they are under continuous change and development (Kuutti 1996). They have a history of their own and the remains of older phases of activities might still be embedded in them as they develop (ibid). Thus, a historical analysis of the activity and of the use of practice in particular is often needed to understand the current situation of the activity (Adams, Edmond et al. 2003). "Existing praxis is historically shaped, and activity theoretical analyses help create links between the past, the present and the future, that are important for ISD" (Bertelsen and Bodker 2000).

The *fourth principle* points out the central role of contradictions as a source of change and development. Activity systems are in constant movement and internally contradictory. "Their systemic contradictions, manifested in disturbances and mundane innovations, offer possibilities for expansive developmental transformations" (Engestrom 2000b).

Such expansive transformations lead to the *fifth principle* that proclaims the possibility of qualitative transformations in activity systems. As contradictions in an activity system are aggravated, some participants may begin to question and deviate from the established norms. Such transformations proceed through stepwise cycles of expansive learning which begin with actions of questioning the existing standard practice, then proceed to actions of analysing its contradictions and modelling a vision for new practices and then to actions of examining and implementing the new model in practice (ibid). An expansive transformation is completed when the object and motive of the activity are re-conceptualised.

The following section provides examples of applying AT in IS research.

### 3.3 AT in IS research

Largely, as a consequence of the significance of Vygotsky's work in development psychology and the theory of learning, AT first made an appearance in the West in the field of educational research, psychology and cognition (Wilson 2006). Education was also the initial field of research of Engestrom, who is probably the best known of the Western interpreters of AT. Engestrom's work however, has moved from education in the narrow sense to the study of learning in work situations (Engestrom 1990; Engestrom 2001; Engestrom 2004) and in the application of technology (Mwanza and Engestrom 2005).

The 1980s-1990s, was a transformational decade for AT, as it came to encompass research topics such as the development and analysis of work activities, as well as the implementation of new cultural tools like technologies (Engestrom, Miettinen et al. 1999). Therefore AT became a promising framework for the IS field (De Freitas and Byrne 2006).

The works of Kuutti (1991) and Bodker (1989) for applying AT in ISD, provided the AT's early impact on the IS field. AT also has a wide audience in other fields of IS research such as the field of human computer interaction (Nardi 1996a) and computer supported cooperative work (Bourguin, Derycke et al. 2001). It has also been used successfully to understand and explain participatory design (Hyysalo and Lehenkari 2001) and collaborative work (Bodker 1991; De Souza and Redmiles 2003). IS analysis and design/development from an AT perspective includes early work by Bodker (1991) on user interface design for work practices and work by Kaptelinin (1996) on human computer interface design and use. Also influential was the work of Korpela, Olufokunbi et al.(1998) on IS across cultures and Bardram's (1998) on the analysis and design of a system in a hospital. Also, important was the AT checklist created by Kaptelinin and Nardi (2006), which provides a detailed guide for the application of AT to the design and evaluation of IS.

In addition to the development of the activity checklist, examples of research relevant to IS include Hasan (2001), who suggested AT to be used in systems development as a guiding framework to analyse the interactions between people and their activities in the IS context and Bodker's (1996) approach to studying computer artefacts in use by analysing videotapes of "breakdowns" and "focus shifts". On the other hand, authors such as Mwanza (2001), De Souza (2003) and De Freitas and Byrne (2006) have suggested the use of AT as a framework for overcoming the limitations of traditional ISD methodologies and therefore they have proposed AT to be used as a software engineering methodology to guide IS developers. Wiredu's (2005) research on applying the theory for studying mobile computer-mediated learning and development presents another recent example of AT in IS. Of relevance to the IS field is also the work of Korpela and colleagues on the work of systems development and the interacting activities of users and developers (Korpela, Soriyan et al. 2000; Korpela, Mursu et al. 2002) and Hasu and colleagues on innovation and development of products, involving the construction of a shared object across the boundaries of user and development teams (Hasu and Engestrom 2000; Miettinen and Hasu 2002).

Activity theoretical ideas are having an increased impact on specific fields of enquiry, such as learning, teaching, human computer interaction, IS etc. However, in all these contexts, AT still tends to appear as an intriguing alternative approach, only partially and briefly revealed to the readers. Its rich texture still remains partially ignored outside Scandinavian IS circles and therefore has not yet become a popular theory in mainstream IS research despite all of its advantages (Engestrom 1999c).

The following section presents the activity theoretical tools that are applied to this study.

### 3.4 The activity system in detail for this study

Vygotsky (1978) has introduced the idea that human beings' interactions with their environment are not direct ones rather they are mediated through the use of tools. Inspired by this concept, Engestrom (1987) extended Vygotsky's original framework so as to incorporate Leont'ev's social and cultural aspects of human activity. Therefore, Engestrom offers the activity triangle model (Figure 3.4), which reflects the collaborative nature of human activity.

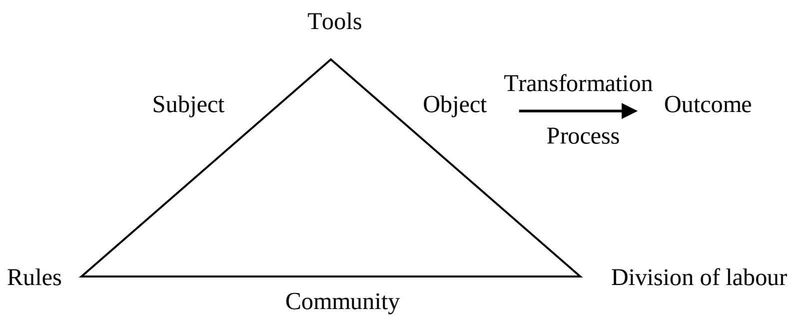


Figure 3.4 Second generation activity triangle model (Engestrom 1987)

Engestrom defined the underlying concept of his framework as that of an:

> "Object oriented, collective and culturally mediated human activity, or activity system. Minimum elements of this system include the object, subject, mediating artefacts (sings and tools), rules, community and division of labour" (Engestrom, Miettinen et al. 1999)

Therefore, Engestrom understood activities as being collective phenomena both with respect to their object (as directed toward an object shared by a community) and with respect to their form (as carried out collectively rather than individually). Activities do not stand alone, rather they exist within a network of activities. The elements of one activity are produced by other activities and the outcome of one activity is needed in one or more other activities (Engestrom 1987).

The activity triangle model represents an outline of the various components of an activity system in a unified whole (Mwanza and Engestrom 2005). Here, the activity itself is the context. An activity is undertaken by human agents (subject) motivated toward the solution of a problem (object) and mediated by tools (e.g. artefacts, methodologies, programming languages, specific applications, practices, cultural means, etc.) in collaboration with others who share the object and who work collectively on its transformation (community). An activity is therefore social within a community and thus influenced by this community. The structure of the activity is constrained by cultural factors including conventions and norms, formal, informal and technical rules that specify acceptable interactions between members of the community (rules) (Barab, Barnett et al. 2002) and social strata, which indicate the continual negotiation for distribution of tasks, power and responsibilities among participants (division of labour) within the context (Mwanza 2001). Particularly, the division of labour can run horizontally as tasks are spread across community members with equal status and vertically as tasks are distributed up and down divisions of power. The activity system itself is continually reconstructing itself since the activity has a dynamic nature which changes and develops during its life cycle (Engestrom 2000b).

The triangular activity system model developed by Engestrom (1987) is considered by several researchers (Kaptelinin 1996; Crawford and Hasan 2006) as a promising framework for the analysis and evaluation of technologies, their development and their use. Created as a tool for describing units of complex mediated social practices, it has relevance to this study as it can identify key aspects of the PP community development reality, point to potential contradictions and provide a visual representation indicating how these aspects are related to each other (Kaptelinin and Nardi 2006). The activity triangle model is indeed a central AT conceptual tool for this study as it allows for creating a rich understanding of the distributed development activity and how this is influenced by the distributed context, the community's rules, norms, history, culture and collaborative practice. This triangular template also allows for the identification of interactions, dynamics and contradictions within the PP context by appreciating the multi-voicedness of the subjects involved and by facilitating the identification of points of tensions as new goals, tools and organisational changes create stress with the current roles, rules and tools (Kaptelinin, Nardi et al. 1999). The section below provides a thorough discussion of the individual components of the triangle model.

### 3.4.1 Elements of activity

The components of an activity system are not static existing in isolation from each other, rather, they are dynamic and continuously interact between them through which they define the activity system as a whole.

***Object and desired outcome***

The key element of an activity system is the object of that activity. The "object is the societal motive of the activity, it defines the activity and separates activities from each other" (Virkkunen and Kuutti 2000). The object's transformation towards some desired outcome or direction motivates the existence of the activity and imputes it with purpose (Hasan 1999). An object can be a physical thing or something less tangible like a plan, or in the case of this study the development of a software system, or it can even be a conceptual object like an idea or theory, as long as it can be shared for manipulation and transformation by the participants of the activity (Jonassen and Rohrer-Murphy 1999). In collaborative activities, such as this study's case of a distributed systems development activity, objects usually arise from negotiation, discourse, or collective reflection (Foot 2001). It is possible, nevertheless, for each individual involved in the collaborative activity to have a slightly different view of the object and purpose of the activity depending on the individual's position in the division of labour, their history in the activity (if they are newcomers or old members), their training and experiences, etc. (Uden 2007). This makes the activity system always "internally heterogeneous and multi-voiced", including various competing and partly conflicting views (Kaptelinin and Nardi 2006). The object of an activity is thus never finite and exactly determined, rather, it can be characterised as a "horizon of possibilities and possible objectives for the actors", something that unfolds in the process of the activity (Virkkunen and Kuutti

2000). This shared object, however, is what drives the activity forward and although some individuals may be more powerful than others in the collective activity, no-one can completely impose their view on others (ibid).

### *Mediated tools*

Another key concept in the activity system is the role of mediated tools. Human activity is always mediated by tools (Bardram 1998). These tools can be a number of things from material artefacts to methodologies and practices. Practices and more particularly the collaborative systems development practices of subjects in their attempt to develop a usable Grid is the key focus of this study and therefore the conceptualisation of such practices as tools that mediate the relationship between subjects and their objective provides useful insights into how development work is performed. In activity theoretical terms, practices are viewed as mediators between the elements of an activity system (Jarzabkowski 2003) by enabling the different elements to interact with each other in the shared practical activity, and so generating continuity. When interactions with other elements and shared activity break down due to contradictions and contested interpretations, practices serve as mediators between the competing views to enable changes in practice (ibid). Practices may be used to mould the context of activity, to “control” new patterns of activity and to re-conceptualise the rationale in which the activity occurs (ibid). As new patterns of activity arise, this may create tensions with the old practices, leading to their modification or alteration. Of course, practices are historically and culturally situated and have a collectively understood status, so there are always likely to be residues of the past in changed patterns of activity (Kuutti 1992b).

As Nardi and Miller (1990) claim, mediated tools distribute thinking, problem solving or other mental actions between the tool and the subject. The mediated tools shape the ways in which people interact with the world and they also reflect the history and the collective experience of people who have tried to solve similar problems before (Barthelmess and Anderson 2002). This experience is reflected in the structures of the mediated tools and in the knowledge of how they should be used (Nardi 1996). The use of mediated tools is argued to be an evolutionary accumulation and transmission of social knowledge. Tools are a reflection of their historical development since they change the process and are changed by the process

(Jonassen and Rohrer-Murphy 1999). Just as activity can be understood by realising the tools that mediate it, the nature of a tool can only be understood in the context of human activity. Tools are developed over time and are local to the community which created them, thus they have a historical and cultural lineage (Turner and Turner 2001). A close examination of tools can therefore offer a deep insight into the history and tradition of the community that are indeed vital elements of the PP community under study. Such examination can provide details as to how their well-known computing history and tradition have shaped their current practice that can in turn provide useful insights into the general GSD practice.

***Community, rules and division of labour***

The last element of an activity system is the community with its rules and division of labour, in which the activity takes place. Activities are socially and contextually bounded; therefore, any activity system can only be described in the context of the community in which it operates. Engestorm (1987) has indeed described the activity as a "system of collaborative human practice" within a community. The community of an activity system "refers to those individuals, groups or both, who share the same general objects and are defined by their division of labour, their rules, and shared norms and expectations" (Barab, Barnett et al. 2002).

The concept of community as it is used in the previous AT literature does not reflect well the complexity of the PP community studied here. Therefore, concepts of community that have been used in the literature and have been described in Chapter 2, such as CoP (Wenger 1998), colonial systems (Porra 1999) and collective mind (Weick and Roberts 1993) and which are directly relevant to this community, are drawn upon in order to elaborate on its complexity. The community, thus, might be a community of practice, a "context within which learning and innovation occurs and is shared" (Venters and Wood 2007), or a colony with a strong history and culture (Porra 2010) or even what Engestrom (2007) defines as "mycorrhizae" – a widely dispersed, fluctuating and weakly bounded community form.

It is argued that the "community negotiates and mediates the rules and customs that describe how the community functions, what it believes and the ways that it supports the different activities" (Jonassen and Rohrer-Murphy 1999). Within the community, individuals might support different activities, and thus formal and informal rules evolve to guide their activities. Assignment of individuals to those activities defines the division of labour, which is also mediated by rules and social negotiation. Individuals, being part of different communities, need to constantly alter their beliefs to adjust to the socially mediated expectations of different groups (Nardi 1996). Conflicts between the different roles in the various communities often arise, leading to transformational activities required to harmonise those contradicting expectations (Barthelmess and Anderson 2002). However, the solid base of the community stands around collaboration.

Having defined the activity triangle model, which is considered to provide a macro-level analysis examining the broader characteristics of the given activity and therefore providing a rich understanding of the Grid's development situation, of the collaboration formed around this development, of the social structures, social processes and problems and their interrelationships, the next step is to define the smaller parts that compose the structure of an activity as well as discuss the concept of contradictions and the framework of expansive learning. These allow a micro-level analysis focusing more on individuals and their interactions, their conflicting ideas, motivations, goals and practices, etc. and particularly emphasise the creation of new knowledge and collaborative systems development practices.

## 3.5 Structure of an activity

Activity in a narrow sense is "a system that has structure, its own internal transitions and transformations, its own development" (Leont'ev 1978). Each activity can be represented as a hierarchical structure organised into three layers – activity, action and operation. The distinction between activity, action and operation became the basis of Leont'ev's three level model of activity (Figure 3.5) emphasising the importance of acknowledging the motivation driving the activity (ibid).

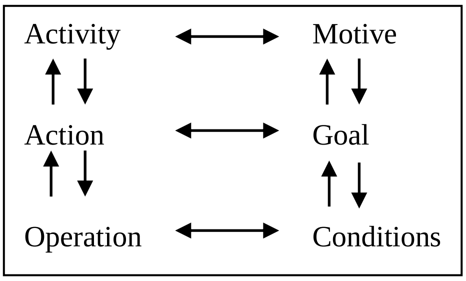


Figure 3.5 The definitive hierarchy of Leont'ev (1981)

The top layer is the activity itself, which is oriented towards a material or ideal object satisfying a need or the subject's expectations. In systems development, an activity could be a software team programming a system for a client. This kind of need state does not explain the direction of activity. Rather, only when a need turns into a motive, a directing force, can a subject find or define their object and the means of obtaining it (Leont'ev 1978). The object in a systems development activity could be the "not-ready-yet" system that should be transformed into a delivered, bug-free application. At the same time however, another activity could exist – the activity of internal project management of the project – where the object is the financial status of the company. The object characterises the motive of activity. Subjects engaged in the activity may/may not be conscious of, or may/may not agree with the collective motive, but it is yet the shared object and its transformation into the jointly produced outcome which define an activity (Kuutti 1996).

Activity is carried out through actions, a sequence of steps or tasks, each of which is not immediately related to the motive, although the sequence as a whole may eventually result in attaining the motive (ibid). For example, the systems development activity described above consists of different planning and problem-solving actions depending on how much code should be produced, how much old code can be used, parts of applications purchased from outside, etc. The actions are controlled by the subjects' conscious goals; these goals are the anticipation of future results of action and are formed and transformed in the collective activity (Turner and Turner 2001). Individuals within a collective activity may have different actions to perform, each with its own sub-goal that combined move the group towards the desired outcome (Adams, Edmond et al. 2003). Actions are usually poly-motivated, therefore, two or more activities can temporarily merge motivating the same action. For example, the action of reporting on the progress of the project will have a different connotation if it belongs to the activity of internal project management rather than if it belongs to the programming activity, although the action and its other ingredients are exactly the same.

The border between actions and activity is blurred. A software project may be an activity for the team members, but the executive manager of a software company may see each of the projects as actions within his or her real activity at the level of the firm. Actions, in their turn, can be broken down into lower-level units, called operations, which are well-defined routines used subconsciously as answers to conditions faced during the performing of the action (Kaptelinin and Nardi 2006). Operations provide an adjustment of an action to the ongoing situation. In a systems development activity, selecting appropriate programming languages, or using operating systems commands could be examples of such operations (Kuutti 1991). People are typically not aware of operations. Operations may emerge as improvisations, as the result of a spontaneous adjustment of an action on the “fly” (Kaptelinin and Nardi 2006).

A dynamic relationship exists between the three levels: activity-actions-operations. Depending on certain experiences such as emotions and feelings of the subject as well as learning, one can be transformed into the other (Leont'ev 1978). Transformation between activities and actions can take place when for example a goal subordinated to another higher-level goal can become a motive, so that a former action acquires the status of an activity. Conscious actions can develop into subconscious operations. An action is usually transformed into an operation when the subject has learned how to perform the action so well that it no longer subordinates their consciousness in performing (Kuutti 1996). A routine operation is usually transformed into an action when it fails to produce the desired outcome and the individual reflects on the reasons for the failure and on how the operation can be fixed. As more of a subject’s actions collapse into operations, the greater the skill development of the subject concerned, leading to a fulfilment of the motive of their activity (Kaptelinin and Nardi 2006).

The phenomena of poly-motivated activities present a special problem in AT. Leont’ev explicitly expressed the idea that an activity can have several motives at the same time (Leont'ev 1978). According to Leont’ev, there can be two types of motives of an activity; the sense-forming motives which give the activity its meaning and the motives-stimuli which provide additional motivation but do not change the meaning of an activity, e.g. they can elicit various behavioural and emotional reactions but they are of secondary importance compared to the sense-forming motives (ibid). Therefore in the case of a conflict, sense-forming motives usually prevail. Leont’ev also proposed a more general conflict resolution mechanism, the so-called hierarchy of motives, which determines the relative importance of the various motives of an individual. But even with this hierarchy the motives that rank higher would determine the course of activities.

Leont'ev's idea has been criticised by many, as not having any practical impact on the fundamental analysis of needs, motives and the object of an activity (Kaptelinin and Nardi 2006). Kaptelinin and Nardi (2006) argue that "activities with several motives are likely to be shaped by the whole configuration of effective motives, not just by one of them"; they can be further influenced by the social context as well as by the conditions that prevail and people's means. They have therefore revised and enhanced Leont'ev's model by proposing a framework (Figure 3.6) that separates the notion of the motive from the notion of the object of an activity. They further explain that this separation implies that if there are several conflicting needs, these needs can correspond to either two different activities or different aspects of the same activity. For example in the case of a systems development activity, the motives behind its initiation can be financial, competing for a job post or even serving a higher shared goal. If the individuals cannot pursue all motives at the same time, then the activity does not have a direction until the object of activity is defined. The object in this case is different from any of the motives, and it is jointly defined by the whole set of motives the individuals strive to attain in their activity (ibid). Moreover, the conflicts that might be created here can be derived from the motives in relation to one another and not from the motives in relation to the object. In other words, each individual motive is articulated in a relatively smooth way to the object, but among themselves, the motives sometimes generate disruptions. Therefore, the object itself is not contested; it is rather the instantiation of the object that may lead to tensions deriving from incompatibilities among the multiple motives (ibid). Of course there are cases that the object itself might be contested, however, objects tend to be relatively long-lived entities.

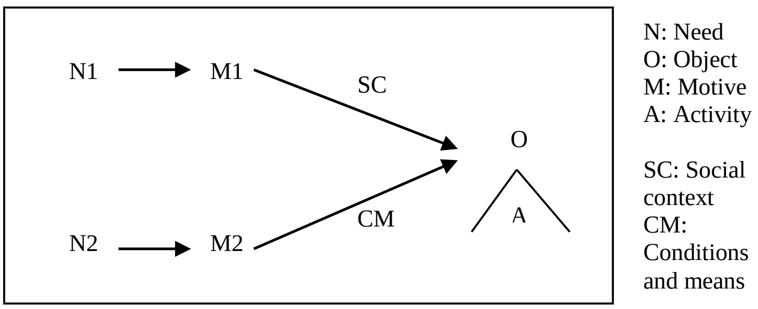


Figure 3.6 Multi-motivated activity model (Kaptelinin and Nardi 2006)

The above model has several distinctive characteristics which differentiate it from Leont'ev's ideas: i) there is only one object of activity, no matter how many motives are involved, ii) the object of activity is jointly determined by all effective motives and iii) the object of activity both motivates and directs the activity.

Leont'ev's (1978) hierarchy of the activity's structure as well as the enhancement of that framework with the multi-motivated activity model (Kaptelinin and Nardi 2006) are relevant and vital analytical tools to this study because they allow the researcher to identify the different interests, needs and motives which made people engage in the activity, the different actions they perform in order to address them and in the end how all these lead to the shared object which is considered to be the only way of achieving what they want.

Leont'ev's tool is employed in this thesis in order to identify such poly-motivation, rather than to provide in-depth interpretations of the activity's structure (activity-actions-operations). Identifying the poly-motivations behind the distributed systems development activity is central to this study since the activity is highly influenced by the different priorities, expectations and cultures of those involved, and therefore such identification allows for the manifestation of contradictions and of the exercise of power (Kaptelinin and Nardi 2006). Understanding the reasons behind contradictions, such as politics and control, etc., is important in this case of a highly political, multi-disciplinary global and distributed systems development undertaking and therefore making this tool extremely useful for this study. Furthermore, the three levels of activity can address what, why and how questions. For example, an activity's motive can answer the why question while the intentional characteristic of goal-oriented actions is a response to what must be achieved. Goals are however achieved in specific conditions that present a problem of how or by what means. Such an analysis therefore can provide a rich understanding and unveil details on how the development activity is performed, emphasising on the systems development methods employed, on the programming languages used, on the testing techniques applied (tools) and on the general computing actions performed.

## 3.6 Contradictions as conceptualizing breakdowns in collaborative activity

Activity systems are characterised by their internal contradictions (Engestrom 1999c). Activities are not isolated units and are therefore influenced by other activities, the collective work and environmental changes. AT provides the concept of contradictions, as a conceptualisation of collaborative breakdowns and tensions. Such contradictions are viewed as highly important in understanding collaborative activity (De Souza and Redmiles 2003), and are relevant to this study of a widely distributed collaborative systems development activity.

Activities are dynamic entities having their roots in earlier activities and bearing the seed of their own successors (Turner and Turner 2001). They are almost always in the process of working through contradictions that subsequently facilitate change (Miettinen and Hasu 2002). Although the object and the motive of an activity give actions coherence and continuity, by virtue of being internally contradictory, they also keep the activity system in constant instability (Engestrom 2000b).

Contradictions emerge from the historically formed objects, means and rules of the participating activities. They usually reveal themselves as breakdowns, disturbances, problems, tensions or misfits between elements of an activity or between activities (De Souza and Redmiles 2003) and identifying them is critical in understanding what motivates particular actions and in understanding the evolution of the activity system (Barab, Barnett et al. 2002). The tool of contradictions plays a vital role in analysing the findings of this study, as it facilitates an understanding of how particle physicists' collaborative practices have developed over the years as well as allowing an appreciation of the major influence their computing history and culture of trust and equality has played in this transformation.

When the components of an activity system begin to misalign, due to internal and external changes e.g. new needs, conflicting priorities and motivations, the activity begins to lose its clear direction. The amount of disturbances and problematic situations increases, leading to dissatisfaction and an increasing need for change (Virkkunen and Kuutti 2000). Subjects will thus try to resolve these contradictions by changing and developing the cultural mediators of the activity, such as the rules of collaboration or by developing new technical solutions and tools (ibid).

Contradictions are considered to be the major source of dynamism, development and learning in AT (Bertelsen 2003), since based on the emerging problems and conflicts, people have to reconsider their position and collectively reconstruct their shared understanding, knowledge and practices. Contradictions, therefore, allow for the renewing of the collective mind itself, for the regaining of the collective capability of acting together and therefore facilitate change (Weick and Roberts 1993).

Engestrom (1987) has provided a description of contradictions through a four-level framework (Figure 3.7).

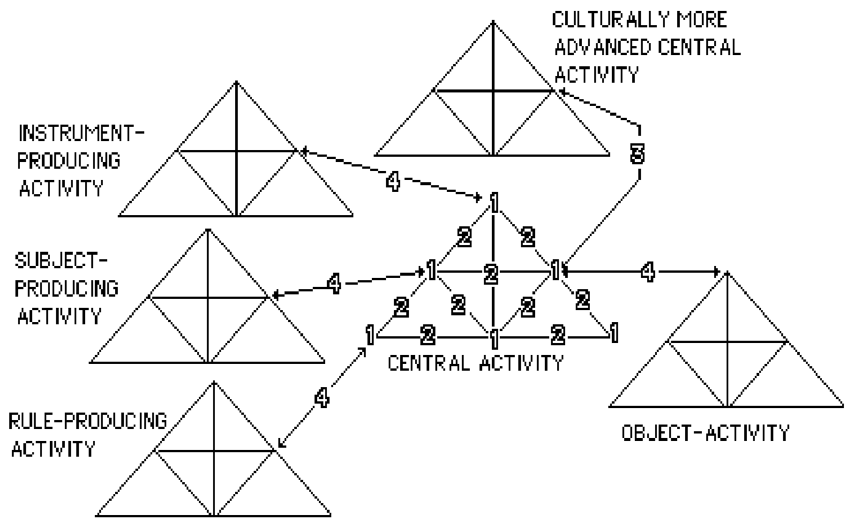


Figure 3.7 Four levels of contradictions within the activity system (Bardram 1998)

The contradictions found within each constituent component of the central activity are described as *primary contradictions.* This contradiction can be understood as a breakdown between actions which realise the activity (Turner and Turner 2001). These actions are usually poly-motivated, e.g. the same action is executed by different people for different reasons, or by the same person as a part of two separate activities, and it is this poly-motivation that may be the root of subsequent contradictions (ibid). In systems development, it is argued that the contradiction between what is said and what is done, e.g. the contradiction between principles, formalisation and specification on the one hand and concrete practice on the other, is a primary contradiction (Bertelsen 2003). The next category of contradictions, are those that occur between the constituents of the central activity and are described as *secondary contradictions.* This for example includes contradictions between the skills of the subject and tools they are using, or between rigid rules and new flexible tools etc. (Engestrom 1999a). In systems development such a contradiction can be presented between the rigid rules of developing a system within a fixed budget and submitting it by a certain deadline with the objective of creating a working system with certain functionalities.

*Tertiary contradictions* may be found when an activity is remodelled to take into account new motives or ways of working. They occur between the object/motive of the dominant form of the central activity and the object/motive of what is called "a culturally more advanced form" of that activity (Bertelsen 2003). A culturally more advanced activity is the one that has arisen from the resolution of contradictions within an existing activity and may involve the creation of new working practices, or artefacts or division of responsibilities, etc. The concept of a culturally more advanced activity does not necessarily imply historical determinism; it can also be interpreted as actual or potential different ways of conducting the central activity (ibid).

Finally, the contradictions occurring between different co-existing and concurrent activities (neighbour activities) are described as *quaternary contradictions* (Barab, Barnett et al. 2002). From this, a complex and continuing evolving web of contradictions may emerge. In systems development, an example of such a quaternary contradiction is the contradiction between the education of computer scientists at the university, focusing on mathematical formalisation and the central activity of computer scientists working as system developers in the industry which is more pragmatic. In conclusion, "primary and secondary contradictions in an activity may give rise to a new activity which in turn spawns a set of tertiary contradictions between it and the original activity and this may be compounded by quaternary contradictions with coexisting activities" (Turner and Turner 2001).

The tool of contradictions is central to this study as it allows the researcher to discover the tensions and breakdowns that arise in the development activity during a period of time, the different solutions that prevail, and eventually the solutions adopted. The next conceptual tool to be discussed is Engestrom's (1999c) expansive learning framework.

### 3.7 Learning as an activity

The dynamic forces of change and learning within the activity are explained by contradictions within and between activity systems (Turner and Turner 2001). A network of linked activity systems is formed across time and space, the links of which are determined by the objects and outcomes of the activities in the network (Virkkunen and Kuutti 2000). Once the object is realised and thus leads to the desired outcome, this outcome will become a part of another activity system; it will become an object to be further transformed in the value chain, a subject, a tool, a rule, etc. (Kuutti 1996). The qualitative changes in these activity systems, caused by contradictions and their resolutions, indicate learning creation.

AT is perhaps "unique among accounts of work in placing such a strong emphasis on the role of individual and group or collaborative learning" (Turner and Turner 2001). Vygotsky's work on developmental learning has been a major influence on the thinking of Engestrom, who has extended the idea to encompass collective learning which he has termed expansive learning (Engestrom 2001) (Figure 3.8). This collective learning is not just aggregations of individuals' learning to perform certain actions; rather the collective learning is different in different phases of the qualitative change in an activity system and in some cases presupposes the creation of innovative new forms of activity and new tools e.g. new systems development practices (ibid).

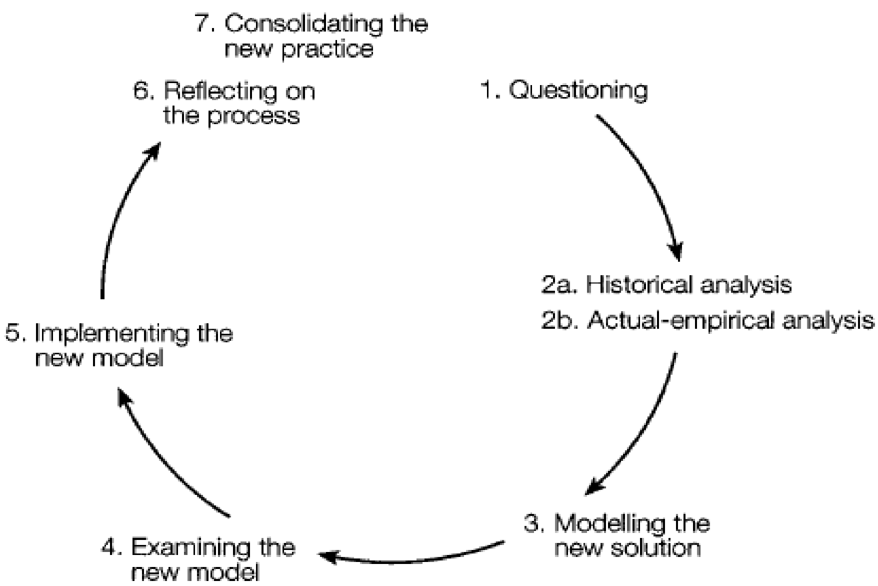


Figure 3.8 Expansive cycle of learning actions (Engestrom 2001)

Expansive learning is about learning what is not yet there. It is the creation of new knowledge and new practices for a newly emerging activity, that is, learning embedded in and constitutive of the qualitative transformation of the entire activity system (Engestrom 2004). The drivers behind these expansive cycles of learning and development are contradictions within and between activities. The expansive learning conceptual tool is relevant and vital to this study since it allows for the identification of new collaborative systems development practices that emerge from the resolution of contradictions to facilitate the distributed systems development of the Grid. Through the identification of these new practices, lessons learned that can form the foundation of a framework of guidance for other communities that deal with GSD are identified.

A crucial triggering action in the expansive learning process is the conflictual *questioning* of the existing standard practice (Engestrom 2000b). An increased discontent and dissatisfaction exists between the subjects that leads to stress and failure. To overcome this situation an *analysis* of the situation and of the problems/contradictions as well as searching for an alternative is required, where new tools, new division of labour and new forms of collaboration are being planned (ibid). The actions of questioning and analysis are aimed at finding and defining problems and contradictions behind them. To overcome the contradictions, the object and the purpose of the activity, what is produced and why, has to be reinterpreted and reconstructed in a wider perspective (Engestrom 2003). If a member of the activity system with a position of power, however, attempts to force a fixed learning assignment in this type of process, it is typically rejected (Engestrom, Engestrom et al. 2002). The third action, called *modelli*ng, involves the formulation of a framework modelling the new solution, the new instruments and the new pattern of activity to the problematic situation (Engestrom 1999a). Understanding the zone of proximal development (Figure 3.9), which is the distance or the area between the present and foreseeable future and represents possible solutions to the problem is important for this phase (ibid).

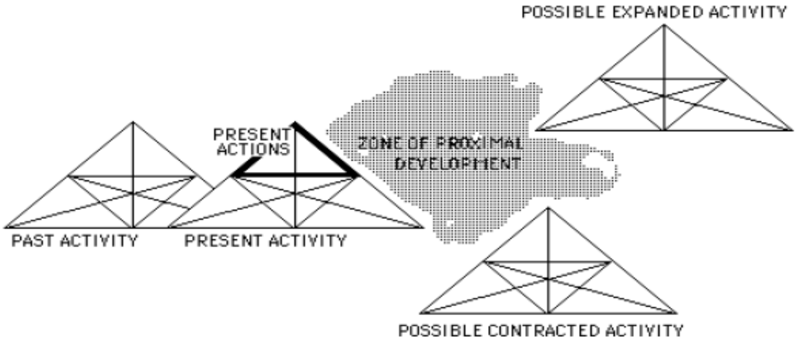


Figure 3.9 The zone of proximal development (Engestrom 1999a)

The change in the activity begins with experiments (*examining*) on the most important parts of the new proposed solution, and then generalises (*implementing)* to the whole activity system. Adopting new tools and practices however might create further contradictions between the old and the new ways of working *(reflecting)* (Virkkunen and Kuutti 2000). This therefore leads to a constant tension between the expansive, future-oriented solutions and the regressive ones that would mean a return to old practices (ibid). Instead of trying to merge the possibly incompatible worlds, a new sideways learning may take place which proposes a solution which gradually gives form to a new practice which may eventually be quite different from the already planned model of the new activity (Engestrom 2001). If these new practices are successful, then they are gradually routinised and new unofficial social practices and forms of cooperation might evolve around them *(consolidating the new practice)* (ibid).

The process of expansive learning should be understood as construction and resolution of successively evolving contradictions in a complex system that includes the object, the tools and the perspectives of the participants (Engestrom 1999c) (Figure 3.10). The phases of a development cycle, do not however, follow each other automatically. On the contrary, “the phases represent possibilities that can be realised through active development work and conscious learning activity” (Virkkunen and Kuutti 2000). Nevertheless, there are cases where the subjects build defences in order not to have to meet the contradictory demands of their situation, or sometimes, the subjects cannot develop a commonly agreed solution to the contradictions and try to manage their situation by individual actions (ibid).

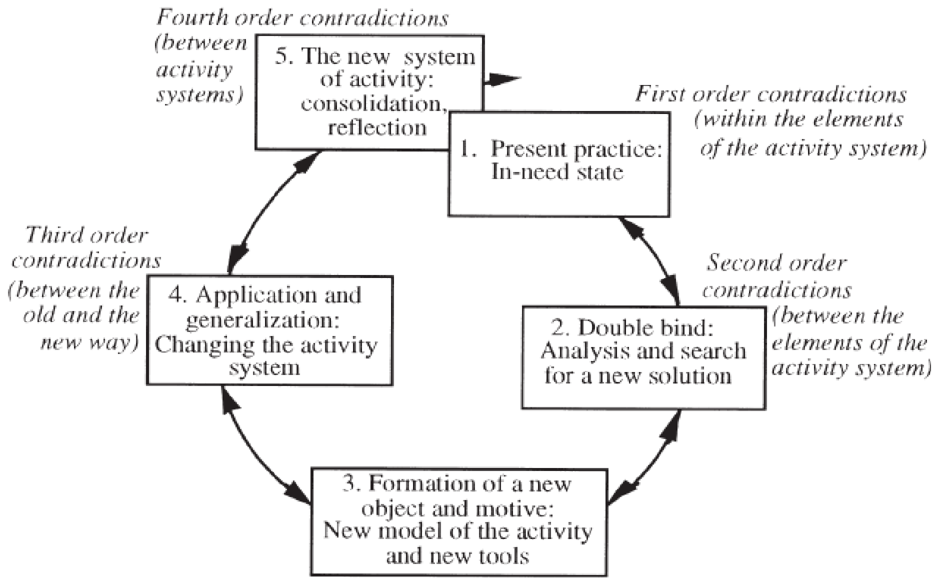


Figure 3.10 The actions of the expansive learning cycle with types of contradictions (Virkkunen and Kuutti 2000)

### 3.8 Chapter summary

This chapter presented a detailed overview of the theoretical framework employed in the research. This research starts with the assumption that systems development beyond the smallest scale is inherently a collaborative activity. AT is a practical theory which enables a rich analysis of complex, changing forms of collaborative human activity (Foot 2001), which is the goal of this study. It provides appropriate conceptualisations and theorising of collaboration as it is focusing on the developmental transformations and dynamics in the collective activity and on how these lead to new knowledge and practices (Bardram 1998). AT has proven to be relevant in situations with a significant historical and cultural context but also in dynamic situations where people, their objects and tools are in a process of rapid and constant change, as in the case explored in this study (Crawford and Hasan 2006). The philosophical expositions of AT have therefore proven to bestow a holistic framework with which to analyse the collaborative systems development practices in the distributed development of the Grid within the PP context. While, this study's AT choice might be argued to underplay the agency of technology within its analysis, with alternatives such as actor network theory (Latour 2005) offering such agency, the complexity of the Grid technology, however, would lead such an approach to detract from the aims of this study and analysis would quickly become lost in the complexities of the technology itself.

There is more to AT than this partial account presented in this chapter, with many different approaches, schools and debates. However, from among the major theorists of AT, Engestrom's approach as the unit of analysis is found to be most appropriate for analysing the activity of distributed systems development within the "exceptional" PP community.

The theory's conceptual tools applied in this study: the activity triangle model (Engestrom 1987); the structure of an activity enhanced with the multi-motivated activity model (Kaptelinin and Nardi 2006); the concept of contradictions (Bertelsen 2003) and the expansive learning cycle (Engestrom 2001) were discussed and their importance to this research was explained. Although the activity triangle model, here used, may look somewhat rigid, it is only for the sake of representational convenience and simplicity (Kuutti 1996). AT does not consider activities as static, rather they are dynamic entities, always changing and developing.

The distributed development of the Grid is a highly collaborative and dynamic enterprise which involves coordinating the actions and interactions of a huge range of people of different culture, education and skill, all working within different projects and institutions around the globe. In summary, therefore, these AT tools, allow the researcher to provide a thorough conceptualisation and analysis of the complexities involved in the collaborative distributed development of the PP Grid.

Particularly, they allow the researcher to (i) develop a rich understanding of the development activity, of how this is influenced by the distributed context, the community's rules, norms and collaborative practice as well as of their sources of tension **(Activity triangle)**; (ii) identify the different interests, priorities, needs and motivations which made people engage in the activity and how their actions lead towards the shared object, or towards conflict leading to misfits and breakdowns in the collaborative work; as well as to understand how individual actions are transformed into a highly collaborative phenomenon which leads to the mutual construction and dissemination of work practices and knowledge **(Structure of**

**activity enhanced with the multi-motivated activity model)**; (iii) discover the tensions, problems, ruptures and breakdowns that may arise in the development activity during a period of time, the different solutions proposed and eventually what solutions are adopted, why (e.g. political reasons, imposed by powerful actors etc.) and how **(Tool of contradictions)**; and finally (iv) identify the new collaborative practices that may be developed to facilitate the distributed development of the Grid, which result from the resolution of the identified contradictions and which can provide important lessons that may be translated into the wider context of GSD **(Tool of expansive learning cycle)**.

Table 3.1 summarises the key AT conceptual ideas and tools applied in this study.

| **AT tools employed in this study** | |
|---|---|
| Activity triangle | • Identification of the **subjects** involved in the activity, their shared **object**, the tools (practices) they employ in order to fulfil that object, the **community** within which they are situated together with its **rules** and **division of labour**. |
| Structure of an activity enhanced with the multi-motivated activity model | • Identification of the different interests, priorities, needs and motivations that made the subjects engage in the activity.<br>• Identification of how their actions driven by different motivations lead towards the shared object, or towards conflict, leading to tensions and contradictions in the collaborative work. |
| Tool of contradictions | • Identification of contradictions**: primary** (within each element of the central activity system), **secondary** (between the elements of the central activity system), **tertiary** (between the object/motive of the central activity system and the object/motive of the culturally more advanced activity system, which can be seen as actual or potential |

| | |
|---|---|
| | ways of conducting the central activity), **quaternary** (between the central activity system and a coexisting and concurrent neighbouring activity system). |
| Tool of expansive learning cycle | • Identification of the subjects' learning cycles in order to resolve their contradictions. The expansive learning process is triggered by **questioning** the standard practice, **analyzing** the situation and the contradictions identified, **modelling** the new solution and **examining** the new model and it is finalised by **implementing** the new model, **reflecting** on the process and **consolidating** the new practice. These phases may not follow each other automatically.<br>• New collaborative practices that result from the resolution of the contradictions are identified here. |

Table 3.1 Overview of AT tools employed in this research

The next chapter provides details of the methodology adopted for this research effort.

# 4 Research methodology

## 4.1 Introduction

In the last chapter, an argument for adopting AT was presented. In that argument, the philosophical assumptions that inform the researcher's position were implied without being explicitly discussed. In this chapter, those assumptions are clarified and a description of the data collection methods and analysis methods used to enact them are provided by discussing the "how" of the entire research. A methodology is a viewpoint, a perspective that expresses the researcher's way of looking at and understanding the world and the phenomenon under investigation. Presenting a detailed methodology and design strategy of a piece of research is crucial in terms of internal validity of its outcomes. This chapter therefore aims at disclosing the methodological decisions made and the justifications underpinning those choices.

Part 1 of the chapter (Section 4.2) provides the overall philosophy adopted in approaching the research, the interpretative philosophy, as well as a justification of this choice for the objectives of this study. This study is also situated amongst existing research traditions in IS. Part 2 (Section 4.3) provides details of the research design; this includes the design strategy followed, the research context, types of evidence, data collections methods and analytical techniques. Justification of all the choices made are presented and related to the objectives of the study and the philosophical approach adopted. Section 4.4 summarises the chapter.

## 4.2 Philosophy of research approach and justification for choice of the interpretative philosophy

The term "philosophy" has been defined as representing human attributes such as beliefs, viewpoints, attitudes, values and way of life (Crotty 2003). The way of life of a researcher is directly associated with their perception of reality (ontology) and its relation to knowledge (epistemology). The philosophy of the research approach is therefore the way in which research is conducted. It is crucial as it underpins all the decisions and choices made in the operationalisation of the research.

The IS field has been described as a "rich tapestry of methodological approaches" (Becker and Niehaves 2007) because of the field's multidisciplinary nature. At the level of the research philosophy paradigm, in Western scientific thought, the two dominating philosophy of approach paradigms are interpretativism and positivism (Myers and David 2002). Recently, critical research has emerged as a third philosophy, but while it has its own distinctive attributes it is usually seen as a quasi-interpretative philosophy, since critical researchers also share an interpretative view of the world (ibid). Traditionally mentioned, the differences between these three research philosophies can be understood in terms of their beliefs about physical and social reality (ontology), knowledge (epistemology and methodology) and the relationship between theory and practice (e.g. how the research is related to the subject under study) (Orlikowski and Baroudi 1991).

This study is founded on the interpretative philosophy of IS research; that is the interpretative ontology and epistemology have guided the researcher's understanding of the phenomenon under study and have informed the data collection. In discussing how the interpretative assumptions are subscribed to the research, this chapter draws upon Orlikowski and Baroudi's work summarising the assumptions underlying an interpretative research philosophy (Orlikowski and Baroudi 2002).

IS is an applied science. Boland and Hirschheim (1987) describe IS as a combination of two primary fields: computer science and management, with a host of supporting disciplines e.g. psychology, sociology, statistics, political science, economics, philosophy and mathematics. In IS therefore, researchers are not only concerned with the design and development of technologies, but also with aspects such as planning, implementation, management, interaction and evaluation of such systems (Myers 1997).Within these attributes, there is an intrinsic nature of the human factor in IS (Avgerou 2000). IS is a social science that relies on understanding based on a social reality consisting of humans' subjective interpretations of technology (Scott 2002). The development and use of technologies in organisations is therefore intrinsically embedded in social contexts, marked by time, locale, politics and culture and influenced by goals, beliefs, values and experiences of individuals and groups as well as by the performance of the technology (Angell and Smithson 1991). Neglecting these influences may therefore reveal an incomplete picture of those phenomena. Interpretive research can help IS researchers to understand human thought and action in social and organisational contexts as it has the potential to produce deep insights into IS phenomena including the management and development of IS (Klein and Myers 1999).

This study explores an IS development phenomenon; the collaborative development of a large-scale and open e-information infrastructure called the Grid. The Grid's development is embedded in the PP context and it is influenced by the culture, norms, ideologies, past experiences, shared goals and beliefs of this community. This research particularly explores the collaborative distributed systems development practices of particle physicists in their attempt to develop and deploy a usable Grid for the LHC, with the aim to portray new and useful systems development practices for GSD. The development effort is loosely coupled, messy, dynamic, distributed and very complex. In exploring such practices therefore, there is a need for a better appreciation of the web of socio-material relationships shaping such Grid development and of the contradictions manifested throughout the development. The exploration of such contradictions is important in this study as it provides evidence of the community's expansive learning and allows for the identification of new knowledge and new practices collaboratively constructed in resolving contradictions which can inform GSD and ISD literature and practice.

Such exploration, however, requires gaining an insight into the collaborative constructed context, revealing information about the history, the structures and the processes of the community undertaking the development activity, and how such context influences the way development work is performed. Particularly, it is necessary to focus on how collaboration occurs, how knowledge is created and shared how competition is minimised and how people interpret and use tools in their everyday work practices. It is indeed anticipated that embedded PP working practices are reflected both in the way the Grid is being developed and in the way the collaborative effort is organised. This research therefore attempts to explain why systems developers act the way they do, by fully capturing the richness of interactions involved in the distributed systems development of the Grid.

Nevertheless, seeking to understand this messiness requires a philosophical base that is grounded in human experiences characterised as a process of interpretation rather than a physical apprehension of the world (Crotty 2003). The access to the study is through interpretations of the subjects involved in the Grid's development who socially construct their reality and knowledge (Walsham 1993) and therefore a paradigm which allows the researcher to focus on the practice in situ – in order not to lose the underlying experiences – and to obtain a more holistic view of the development activity is required (Keen 1991). The perspective of the researcher here, is of one who is "inside" the group, observing and interpreting what is happening (Trauth and Jessup 2000), and therefore a paradigm that focuses on human sense-making as the situation emerges is needed.

The nature of this study does not favour a positivist philosophy of testing hypotheses, calculation and formalisation (Crotty 2003). This study is based on unveiling the subjects' interpretations concerning the Grid's development by exploring their complex social interactions and the meaning they assign to their practices. The practices employed or constructed during the development of the Grid cannot be understood independent of these social actors and they gain importance by being socially shaped and changed through the subjects' actions and interactions and through the meaning and beliefs (including culture and history) they attach to them (Klein and Myers 1999). This study thus rejects positivistic ideas of a reality and knowledge being objectively given and described only by measurable properties (Bryman 2004). In this study, beliefs, experiences, history, culture and interpretations all influence the development work, the knowledge being constructed and the practices employed and emerged. This translates into the idea that knowledge cannot be based on objective facts or predefined variables, it cannot be derived from deductive proof and the empirical testability of theories and facts, and subjectivity cannot be ruled out completely, all ideas encompassed by the positivist paradigm (Hirschheim 1985). While positivism has achieved significant research success over the years, particularly in the natural ("hard") sciences, where the phenomena under investigation were not hard but soft and were related to historical and contextual conditions as possible triggers of events in human action, positivism was indeed found not to be successful (Orlikowski and Baroudi 2002).

On the other hand, although this research's study of contradictions could suggest taking a critical theory perspective, the nature of this research does not favour a critical theory philosophy of understanding and critiquing the material conditions of power and domination, thereby initiating change (Ngwenyama 2002). Critical theory involves "value-laden preconceptions and emotionally loaded political and moral stances" (Avgerou 2005), which are essential elements helping to improve our understanding around issues related to IS and social change in contemporary society. An important distinction of critical studies is its evaluative dimension. While such critical philosophy attempts to critically evaluate and transform the social reality under investigation though longitudinal case studies or ethnographies by critiquing existing social systems and revealing contradictions and conflicts that may exist within their structures (Orlikowski and Baroudi 2002), once these contradictions are discovered, the researchers have a duty to inform the participants of their condition and attempt to transform the social reality under investigation by proposing plans of action to remove the source of the alienation (ibid). An important objective of the critical philosophy is thus to create awareness of the various forms of domination, so that people can act or attempt to eliminate them (Avgerou 2005). The goal of this research, however, is not to critically evaluate particle physicists' systems development practices and therefore offer possibilities for reconstructing them or to discuss power issues in the community and offer solutions on how to eliminate them. Rather, the researcher's aim is to document a plausible coherent story of what is happening throughout the Grid's development and therefore aim to understand and unveil the underlying structure of the systems development phenomenon under study. In particular by focusing on the PP community's tools, rules, division of labour, culture and history, and to use this knowledge to 'inform' other settings.

A number of studies in ISD within organisations have indicated that an interpretative approach to facilitate the research of the systems development process is the most appropriate vehicle for the study of this phenomenon (Butler and Fitzgerald 1997; Hughes and Wood-Harper 1999; Berntsen, Sampson et al. 2004). Interpretative studies involve understanding such a phenomenon subjectively within cultural and contextual situations and from the perspective of the participants through the meanings they assign to it, allowing us to capture "richer" information about the situation (Walsham 2006). The interpretative tradition recognises that meanings are formed, negotiated and collectively reconstructed and therefore the interpretations of reality may shift over time (Orlikowski and Baroudi 2002) allowing for the identification of contradictions in social relations, something which is important in this study. Meaning and intentional descriptions are central in the interpretative philosophy because they reveal the subjects' points of view and constitute their behaviour, since they are constructed by subjects while engaging with the world (Nandhakumar and Jones 1997). Organisational structures and collaborative social relations, therefore, are not seen to be objectively unproblematic and conflicts can arise. Interpretative assumptions allow for the identification of conflicts without the researcher having to explicitly propose plans of actions on how to eliminate such contradictions unlike in the critical philosophy (Orlikowski and Baroudi 1991). The interpretive philosophy allows for the explanation of phenomena as embedded in the social sphere and the relevance and validity of such interpretative research rests in accounting for these multiple meanings (Kaplan and Maxwell 1994). Indeed, interpretative researchers, aim to come up with plausible coherent stories and rich descriptions of what is happening and use this knowledge to 'inform' other settings, rather than to predict or provide canonical rules (Checkland and Scholes 1990). This is the goal of this research; hence, although generalisation, in the form of providing rich descriptions (Lee and Baskerville 2003) is taken into account and exhibits the usefulness and the potential summative validity (Lee and Hubona 2009) of this research, it is still up to other researchers to appropriate the findings and reflect on their practice in light of them.

Drawing on the work of Walsham and Waema (1994), interpretative assumptions in IS research which are aimed at producing an understanding of the context of the IS, found to be consistent with the nature of this study. The knowledge in the PP community is considered subjective and the construction of such knowledge is subject to the individuals' interpretations and is constantly revised when changes are created in their context (Walsham 1995). The adopted interpretative research methods of this study can therefore provide deep insights into the structure, history and processes/practices involved in the Grid's development (Walsham 2006). As Orlikowski and Baroudi (2002) argue, "the aim of interpretative research is to understand how members of a social group, through their participation in a social process, enact their particular realities and endow them with meaning, and to show how these meanings, beliefs and intentions of the members help to constitute their social action", which is what the researcher attempts to achieve here.

Despite the benefits of interpretative research, several concerns have been raised regarding its suitability to IS research as well as regarding how interpretive field research should be conducted and how its quality can be assessed, with some scholars arguing for the need to be more critically informed and to better understand the inherent struggles and sources of alienation within society or an organisation (Walsham 2001; Avgerou 2002). Another point of criticism was raised by (Nandhakumar and Jones 1997) who identified limitations that might endanger the research quality. These are: 1) the ability of the researcher to understand the actor's interpretation correctly, especially if both are from different social and cultural conditions, 2) the problem of the difference between what is said and what is done, 3) the issue of secrecy in social interaction and finally 4) actors themselves might not be able to give an account of their action since those behaviours form part of their social routine of which they may not be fully aware. However, Klein and Myers (1999) have stressed that such limitations can be minimised by creating a set of principles guiding interpretative research, although these should not be seen or treated as a pre-determined set of criteria, rather as a framework of guidance which can lead to high quality interpretative studies. These are: the principle of the hermeneutic cycle, the principle of contextualisation, the principle of interaction between researchers and subjects, the principle of abstraction and generalisation, the principle of dialogical reasoning, the principle of multiple interpretations and finally the principle of suspicion.

The proposed principles have been applied to validate this study in order to make this interpretative research more plausible and rigorous (Klein and Myers 1999). A critical reflection of the social and historical background of the research setting was performed through interviews and secondary sources unveiling the distinctive nature, culture and history of the PP community as well as the contradictions that emerged from early Grid development and how these led to the collaborative systems development practices employed today. Data collection and interpretation were informed by hermeneutics (Butler 1998). The findings were thoroughly discussed with participants and their analysis is the result of iterative reflections with the researcher and members involved in the Grid's development, therefore enabling a triangulation of the data. Differences in interpretations among the participants were also taken into account and were resolved by crosschecking the findings with participant observations and other secondary material. As the purpose of an interpretative study is to understand the meaning of the "real world" though interpretations of individuals (Guba and Lincoln 1994), this type of study is value-laden and therefore the existence of bias is recognised. Such bias was, however, minimised by collecting data from multiple sources. Finally, although generalisation from the study to other contexts is not the main objective of this research, generalisation was still taken into account and a framework of guidance towards engendering HEAD_SDC was created so that other organisations or communities can reflect on their own practice and perhaps foster such a similarly dynamic flexible community in order to manage their global development efforts.

Together the ontological and epistemological assumptions of interpretativism have provided the basis for understanding the key issues as well as for theory development in this study. The philosophical assumptions of AT match with those of interpretative research. In undertaking this research the researcher was careful not to base their understanding of global systems development practices on existing formulations and conceptualisations; rather the data collected were understood within their context of emergence. This means that weight was placed on how the subjects' perceptions, norms, culture and history shaped their understandings, motives and goals within the conditions in which they performed computing actions.

The following section provides details on the data collection and analysis methods employed in this study.

## 4.3 Research design

A research design provides the framework for the collection and analysis of data (Bryman 2004). A choice of research design reflects decisions about the priority being given to a range of dimensions of the research process. It is considered to be the structure of any scientific work as well as to provide direction to the research. In the following sections the research design adopted for this study is presented.

### 4.3.1 Case study strategy

Based on the aims of this research and in relation to the interpretative philosophy adopted and explained above, it was important to adopt an investigative strategy that would facilitate the exploitation of sufficient and rich insight of the empirical phenomenon. A strategy that allowed for the examination of the problem in its real time “unique” PP context was therefore needed. The LCG collaboration is unique, preventing comparative studies, rather providing a revelatory case of distributed systems development practice. Hence the choice of an interpretative case study was found to be the most useful to seek an in-depth understanding of the dynamic, complex, loosely coupled particle physicists’ systems development activity and the collaborative construction of their shared practices. In approaching this study it was evident that particle physicists’ intensive ongoing communication had shaped their work. As Knorr-Cetina (1999) highlights “Discourse runs through high energy physics (HEP) experiments; it provides experiments with a massive spectacle of object features, of their story lines, and technical dramas, which are held by and spill from computer displays and printouts, transparencies, internal notes ‘documents’ and together with all these, talk…[ ]. Discourse, channels individual knowledge into an experiment, providing it with a sort of distributed cognition or stream of collective self-knowledge which flows from the astonishing intricate webs of communication pathways”. This highlighted the need for a research method that captured the ongoing dramas and discourse of participants. The choice of a case study was found to be appropriate in revealing a rich understanding of social phenomenon and capturing the ongoing interactions of participants within the collaborative context and how these were shaped by the context.

A case study is defined by Yin (2003) as “an empirical enquiry that investigates contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident”. Similarly, Benbasat, Goldstein et al. (1987) suggest that a case strategy is an appropriate way to research an area in which few previous studies have been carried out, while Bonoma (1985) proposes case study research for phenomena that involve sticky practice-based issues where the experiences of the actors and the context of actions are critical. The messy, complex and multidisciplinary distributed systems development activity of the distributed and open Grid infrastructure within the PP context represents a contemporary phenomenon, where few previous studies have been conducted and in which the boundaries are blurred and research and theory are in their formative stages. The problem of this study thus exhibits the features suggesting undertaking a case study research design – real-life, contemporary with blurry boundaries and practice-based – and that is why the interpretative case study design strategy was adopted.

A number of authors have stressed the usefulness of case study research as a method applied in IS phenomena, with most of them discussing its ability to provide a rich and detailed picture of the situation within a limited period of time (Bell 1993), to capture sufficient knowledge of practitioners for generating theoretical frameworks from the field work (Benbasat, Goldstein et al. 1987) as well as to facilitate a solid understanding of the nature and complexity of the context under study (Walsham 1995; Creswell 1998), making this strategy suitable to address the objectives of this study. The aim of this study, as discussed above, is to explore the interactions between participants and understand the meaning they assign to their work practices in an attempt to describe their "expansive learning" process which gives rise to new methods, tools and work practices. Indeed, the case study research design allowed the researcher to study the development phenomenon in its natural setting by delving into the complexity of the context (Benbasat, Goldstein et al. 1987) – which was particularly important in the PP context where complexity and uncertainty prevailed – while maintaining the holistic and meaningful characteristics of real-life events such as the organisational and managerial processes involved, the systems development processes followed, the relations and interactions between the subjects, etc. (Orlikowski and Baroudi 2002). This research has undertaken an exploratory in Yin's (2003) interpretation, or an intrinsic in Stake's (1995) definition case study.

The next section provides details of the research context and explains its significance.

### 4.3.2 Research context

Grid technology is claimed to be a fundamental step towards the realisation of a common *service-oriented infrastructure* for on-demand, distributed, collaborative computing, based on open standards and open software (Foster, Kesselman et al. 2003), promising to enable the sharing of computer processing power, storage space and information on a global scale (Berman, Geoffrey et al. 2003). Interestingly, Chatterjee (2002) described the Grid as the "foundation of the 21st Century information infrastructure". While this is obviously a bold statement and interest in

Grids *per se* has waned compared to Cloud computing with its more business focus (Bandyopadhyay, Marston et al. 2009; Hey, Tansley et al. 2009), yet Grids focused on globally distributed resource sharing (Foster and Kesselman 2004) still remain an important step towards global IT infrastructures and remain relevant particularly for high performance scientific research where large-scale resource-sharing is necessary such as in PP (Kyriakidou and Venters 2009), geophysics, earth sciences and in industries such as financial services, engineering (Wouters and Schroder 2003) and health (Jirotka, Procter et al. 2005).

Grid concepts remain central within Cloud computing efforts (Armbrust, Griffith et al. 2010) as the systems involved are similar (Abbot 2009) and research in Grids remains relevant to their technical challenges. Further, "it is anticipated that as Cloud computing will promote collaboration among different stakeholders to develop complex IT infrastructure so Cloud will develop towards including multiple administrative domains" (Willcocks, Venters et al. 2011). Understanding collaborative Grid development therefore will prove relevant to such endeavours.

A Grid is a large number of globally distributed processors and other computing devices, which are linked through fast networks, span multiple administrative domains with no single authority in charge and are presented to the user as a single computer and without the need to address individual resources directly. Among many international Grid projects worldwide, PP stands out, because of its exceptionally distributed collaboration (Chompalov, Genuth et al. 2002), its significant contribution to the Grid's development and the fit of its style of analysis to the Grid's capabilities. Being the first scientific community to be involved in the development of such a large-scale infrastructure, its contribution can be influential in the way other communities develop, adopt and conceptualise large-scale systems in general and the Grid in particular.

The PP Grid and the PP community provide a relevant case for studying distributed systems development for a number of reasons. Firstly, it is part of a wider initiative, the e-Science vision, which aims to produce a new generation of computing infrastructures to support collaborative and distributed working in science (Hey and Trefethen 2008). Secondly, this Grid is being developed in a distributed collaboration on a large-scale with many independent participants, rather than by one or a few innovative companies. This is not surprising, since particle physicists have for many decades worked as a globally distributed community that thrives on democratic debates and discussions (Knorr-Cetina 1999). They are a community of distinctive history, culture and practices and their collaboration has been described as "exceptional" (Chompalov, Genuth et al. 2002), something which is expected to influence the systems development practices they employ. Thus, it is anticipated that embedded PP practices come to be reflected both in the way this Grid technology is being developed and in the way the collaborative effort is organised.

It is claimed that PP has, for a long time, exhibited an ability to assimilate new high performance computing resources within its work practices in a successful and highly pragmatic way (Knorr-Cetina 1999). Indeed, it is argued that the World Wide Web grew out of the need of particle physicists to share data (Berners Lee and Fischetti 1997). The technological transition from supercomputers, to the web, to open source server farms and lately to the early development of the Grid, has been claimed to be centred on their need for doing physics (Anderson 2008). In doing this, it has been suggested that they have illustrated their ability to work pragmatically as one of the biggest globally distributed communities (Knorr-Cetina 1999). The nature of experimental PP work is also argued to be quite distinctive. As Venters and Cornford (2006) argue, "the experiments that underlie the field require very large capital investment beyond the possible budgets of most individual national science programs, or their stocks of human resources. Due to their cutting-edge nature and their complexity they also have very long lead-in periods on their construction, followed by long periods of operation".

Interestingly, Knorr-Cetina (1999) argues that such experiments are achieved within organisational structures which have very few formal lines of authority, working as global virtual collaborations of different people with different (though broadly aligned) aims. It is therefore surprising that experiments of this size work at all, but as Knorr-Cetina (1999) claims they work, by scientific standards, successfully. Knorr-Cetina (1999) indeed states that sociologically it is more surprising that the ones observed work in somewhat non-bureaucratic ways, without overbearing formal organisation, without hard-set internal rules and without the management problems apparent in industrial organisations of comparable size. As Shrum, Genuth et al. (2007) suggest, the PP community does not organise workforce of employees in industrial-like ways, but they bring about truly collective forms of working by enticing participants into some form of successful cooperation. This communitarian form of working is what is argued to make them unique and interesting (Knorr-Cetina 1999).

Whereas the LCG project within the PP context is the central focus of the research and the point of reference, it was necessary to take into account other people involved in the development. Developing the Grid is a novel distributed effort, requiring a number of people from other sciences to be involved. Funding, manpower and resource requirements meant that other projects, such as country-focused projects, e.g. GridPP (UK) and INFNGrid (Italy) etc., as well as other European or international projects, such as EGEE were involved in this Grid development, contributing towards the LCG. The context for this research thus encompassed the combined development activity of all those involved. This included people from a range of disciplines, including particle physicists and computer scientists. As the findings section will demonstrate, however, the PP culture is still dominant, strongly influencing the development work as well as providing direction and vision. Given the resource limitations, funds and practicality, it was unfeasible to cover all elements of the vast distributed LCG project and this made it necessary to focus on the UK and CERN in particular, therefore much of the research is based on the LCG, EGEE and GridPP organisational contexts, which also happened to be key projects involved in this development effort. Yet, the boundaries in this context are flexible and thus people are able to work through them, therefore, as many interviewed were involved in a wide variety of roles in various projects – some extending beyond LCG, GridPP and EGEE – making the research coverage sufficient to provide a rigorous account.

### 4.3.3 Data collection

Given the aim to understand the particle physicists' systems development activity through the examination of their structures, working practices, tools and collaborative history and culture, the study relied heavily on qualitative data. Various means of data collection were employed including interviews, documents and participant observations to collect relevant qualitative data (Bryman 2004). Qualitative evidence grounded the researcher's understanding of the particle physicists' development activity in their social and cultural context in which they learn, something which according to Kaplan and Maxwel (1994) would have been largely lost with the quantification of textual data.

This research was part of and sponsored by the Pegasus project (EPSRC: Grant No: EP/D049954/1) and therefore the researcher was part of a research team. The data collected and presented in this thesis were collected by the researcher and by the rest of the team. The researcher conducted 43 interviews out of a total of 82, while the rest of the team conducted the remainder. All interviews were coded and analysed, however, by the researcher herself and therefore the research presented here is based on the researcher's interpretations of the situation.

The main field work for this study was conducted between November 2006 and March 2011, and took place in three main phases. The people interviewed were carefully selected so as to encounter a diverse set of actors with different roles working in different institutions. The people involved in this large-scale development usually had different roles in the different projects and therefore their contribution to this research was not restricted to a single area. Representative people were interviewed from among middleware developers (both particle physicists and computer scientists), users from all four experiments, the requirements gathering team, the deployment and integration teams, system administrators, computing coordinators, team leaders, project leaders, and of the EGEE, GridPP and LCG management. A detailed list of people interviewed is presented in Table 4.1.

The frequency of the meetings and interviewing activity was high (see Appendix B for more details). During these years of recursive field work the researcher attended major collaboration meetings and workshops usually lasting 4 days to 1 week, such as the GridPP collaboration meetings, worldwide Large Hadron Collider computing Grid (WLCG) collaboration workshops and All Hands meetings, which gave the researcher the opportunity to gain a rich understanding and insight into how this diverse community works and collaborates. The community welcomed the researcher and appreciated the fact that there was an interest in describing their systems development practice, since this was something that would help them to further develop their practices. The opportunity for constant interaction with the community allowed intensive reflection during the process of research and was essential in gathering knowledge and insight information about the history/culture, processes and structures involved in the development of the Grid with special consideration for the difficulties, conflicts and perceived challenges.

| **Participants involved** | **Number of interviews** |
|---|---|
| Middleware developers – GridPP, LCG, EGEE (including developers who write applications to run on top of the middleware) | 17 |
| Deployment team | 8 |
| Integration team | 5 |
| Pre-production team | 3 |
| Tier-1 and Tier-2 coordinators | 5 |
| Systems administrators | 6 |
| EGEE and LCG (managerial site) | 8 |
| Users (ATLAS, CMS, ALICE, LHCb) | 11 |
| Computing coordinators (experiments) | 4 |
| GridPP Project management members | 10 |
| Requirements gathering | 5 |
| ***Total*** | ***82*** |

Table 4.1 Summary of interviews conducted

The first phase of the field work was initiated with the researcher attending the 17$^{th}$ GridPP collaboration meeting (November 2006) which provided important insights into the way physicists collaborate to solve their problems and coordinate their actions. A short pilot study followed in March 2007, which was conducted at CERN and consisted of semi-structured interviews with representatives from both the developer and the user groups within the PP community, friendly informal conversations as well as on secondary data such as documents and wikis. Prior to every interview, the interviewees were briefly informed about the aims and the nature of this study. The interviewing was based on a set of basic themes and an evolving list of open-ended questions. In general the questions asked were guided by the answers of the interviewees. The pilot study's purpose was to create a rich understanding of the LCG project, identify other important sub-projects involved (e.g. GridPP, EGEE) and key participants and understand the collaborative nature of the community. This study was a preparatory one, which made the researcher known to the community. The first phase was completed with another round of semi-structured interviews with middleware developers at Rutherford Appleton Laboratory (RAL) (UK) on December 2007 and February 2008. Together with the pilot study already undertaken, these formed the initial findings for the elaboration of a more detailed interview protocol, which led to a more in-depth exploration during the second phase.

The second phase was initiated with interviews at RAL (February-March 2008) and 2 weeks spent at CERN on April 2008. It continued with semi-structured interviews and observations of key participants at universities across the UK between March 2008 and November 2008. The second phase was primarily based on semi-structured interviews as well as the observation of the developers' daily activities. The focus was on (i) the distributed systems development activity physicists (and computer scientists) were undertaking, including methodologies/practices they used, programming languages and environments, collaborative tools that supported the distributed development, requirements' gathering, kinds of testing applied, integration of distributed components and in general the evolution of their practice; on (ii) how they collaborated in order to promote a sense of standardisation, plan their actions, communicate results and know-how, solve technical problems, etc.; on (iii) the unique community bonds they created and how these shaped the way they worked and finally on (iv) the identification of any tensions and contradictions that were created and how these were resolved. The second phase, in addition to following current development work, involved substantial data gathering and analysis to understand the collaborative history and culture of the PP community, which plays a significant role in shaping the way particle physicists work and the practices and tools they develop and employ.

The third phase was established and performed for the need to finalise some issues and ideas. A final round of interviews/informal discussions was performed on November 2009 and March 2011 with key people in the project, in "leading" positions. The researcher was able to finalise some concepts and ideas that came out from the two previous rounds of fieldwork, discuss research findings and obtain feedback.

A detailed description of the research activity, including people, dates and places of interviews and participant observations is presented in Appendix B. The interview protocol is also presented in Appendix C.

### 4.3.4 Data sources

Research methods that captured the ongoing interactions of participants within the collaborative context were used. Semi-structured interviews, informal conversations, participant observations and the attendance of meetings/workshops were the primary data sources, with documents, archival data, websites/wikis, and emails as supplementary sources. The above data collection methods are discussed in detail:

***Interviews, informal conversations***

At the core of this research are more than 80 semi-structured qualitative interviews of between 1 and 2 hours, undertaken at various universities across the UK and during 2-week periods at CERN in Geneva. Interviews were audio-recorded, transcribed and then coded using the ATAS.ti software to derive themes and concepts. However, for the most part of the analysis reported here, is the result of the interactions amongst the researcher and other people involved in the Grid's development, rather than a narrow machine derived account.

Apart from on-site planned interviews and observations, unplanned meetings and discussions happened *in loco*, providing opportunities for more interactive information gathering. At times, discussions were carried out during lunch or dinner time and in the pub, or conversations continued along the corridors in a more relaxed and friendly way.

***Participant observations, meetings and workshops***

Participant observation was utilised as a data collection method in order to obtain a rich understanding of the context, the history and culture, the structure of the project and of the system being built.

The researcher, during one of the trips to RAL and CERN, had the opportunity to make two observations of developers' daily activities and therefore to explore the systems development activity itself, focusing on its complexity, messiness and the means employed in coping. Specific focus was given to the methods and tools employed and the challenges faced during the development. Two additional observations of users' activities were also performed, which helped the researcher to gain an insight into how the Grid works and what sort of applications should run on top of the middleware, but also allowed for an understanding of why particle physicists develop their own technological solutions.

The researcher attended almost all major meetings and workshops performed, during the 2 years of intensive field work. This contributed to the creation of a more relaxed environment between her and the community's members, which enabled her to audio- and sometimes video-record the meetings. Attending and observing their formal and informal meetings and workshops was important in understanding the PP community as the product of a long history, and this history influencing the current community and the way it worked. Furthermore, it allowed the researcher to comprehend how particle physicists work, plan their actions, deal with problems, offer solutions, promote best practice techniques, share expertise and knowledge, and create and share visions and strategies. The WLCG workshops, however, provided the opportunity to appreciate the magnitude of this community and of this Grid development effort. People from all around the world attended these workshops and it was an opportunity to understand how this whole distributed effort was organised and guided towards one direction and one high-level goal, the need of "doing" physics.

An electronic diary was kept by recording observations of a wide range of practices enacted by the community's members as well as about interesting occurrences. The diary consisted of Microsoft word documents, usually created for a specific workday or meeting. The researcher consistently kept notes during the observations and later transferred these to the electronic diary. The diary was organised thematically rather than in a chronological order allowing for easy review of interesting themes and further development of a previously recorded observation from a different point of view, or in the light of a new observation.

***Documents, emails, websites, wikis***

Formal documents that described the vision, mission and strategy of the community under study were collected and examined. These documents were typically electronic files discovered on the projects' websites, or provided by a number of people involved in the project, such as CERN computer letters, bulletins, formal reports, newspaper articles, member's publications, EGEE and GridPP project's material, etc. The documents included software development codes of practice, presentation materials, quarterly reports, funding reports, archival records including organisational charts, plans, etc.

It is believed that these documents were carefully engineered texts that had been designed to circulate ideas that reflected the reigning orthodoxy. The researcher realised that formal documents said little about the process by which beliefs about systems development and systems development practices were established as being legitimate. Therefore, informal documents collection proved to be more analytical and meaningful and provided more valid data. These informal documents included but were not limited to, minutes of meetings, memoranda, emails, access to wikis, etc. Informal documents often expressed alternative ideas and provided insights into how the legitimacy of some beliefs had been established. Furthermore, following discussions on their emails and wikis proved to be significant since interaction through these means was more frequent than their actual face-to-face meetings and therefore provided an understanding of the more technical aspects of the project, of how problems were being solved and how directions were given as well as indicated the willingness of the community's members to help each other.

These secondary sources complemented the other data sources to build some unifying and holistic evidence. More importantly, documented information from such secondary sources enabled the researcher to examine the evolution of ideas before beginning the study and therefore informed the design of some of the interview questions.

Table 4.2 provides details of the research activities undertaken.

| Research methods | | Examples | Data collection |
|---|---|---|---|
| Semi-structured interviews | | Members of GridPP, LCG and EGEE, middleware developers, users, system administrators, technical coordinators… | Audio-recorded, transcribed, coded |
| Participant observations | Face-to-face meetings | GridPP collaboration meetings and deployment meetings, WLCG workshops, All hands meetings … | Audio-recorded, some video-recorded, notes taken, not transcribed |
| | Developers' and users' daily activities | Middleware developers (particle physicists and computer scientists) at RAL and CERN, CMS user at Imperial, ALICE user at CERN | Some audio-recorded, some video-recorded, notes taken, not transcribed |
| | Site visits | GridPP sites, experiment visits at CERN... | Notes taken |
| Secondary data | | Publications, formal documents, reports, websites, wikis, emails… | Frequent consultations |

Table 4.2 Details of research activities

### 4.3.5 Approach to data interpretation and analysis

The data collected from both primary and secondary sources were characterised by texts and symbols including voice recordings, interview transcriptions, notes and pictures. In order to satisfy the aim of this enquiry, data collection and interpretation were informed by hermeneutics, a philosophical theory of interpretation (Crotty 2003). Hermeneutics is concerned with the interpretation of written texts and human understanding on how a reader can correctly interpret the meaning of a text written by another individual that comes from different social and cultural backgrounds. In hermeneutics, texts are perceived as the media that transmit experience, beliefs, values and judgments from the author to the interpreting subject (ibid).

In an attempt to correctly interpret the project's texts, the methodological principle of the "hermeneutic circle" was employed (Rathswohl 1991). The central idea of the hermeneutic circle is "learning the whole through learning the part", which involves the process of interpreting/reinterpreting back and forth between the part and the whole. Repeating this process enabled new interpretation and refinement of presuppositions, which in turn helped to capture a true meaning of the whole.

In this research, the interpretation and analysis of the texts were tasks partly conducted during the collection process. The hermeneutic process started with the researcher's early understanding of the development of the Grid as a distributed collaborative activity based on information gathered from the pilot study. Through the interviews, observations and documented information collected during the main field work, the researcher was able to refine her understanding and address the research questions. Where conflicting or incomplete interpretations existed, the researcher would go back to participants asking for clarifications. This hermeneutic process ended when it was felt that a proper understanding of the distributed systems development activity under study was reached.

Two stages of data analysis can also be identified. The first round involved open coding of the data (using the ATLAS.ti software), labelling aspects of the project, their practices, and emerging ideas from the phenomena. Being familiar with the texts collected was necessary in identifying the relevant themes and providing correct interpretations of what was going on. This exercise, combined with the embedded understanding acquired by the researcher during secondary material research and participant observation, provided an appreciation of the complexity of the project and the tensions inherent in such work. For example, physicists being experimental in nature are not keen to follow any given methodologies; rather being powerful computing experts they prefer to think their way forward through trial and error. Also, while there may be seem to be a lot of complexity, instability and a constant negotiation for dealing with problems in the project, they all have the confidence that the "Grid will work". Furthermore, the LCG community was not unified in their opinions. Tensions, conflicts and different views existed and are of course inevitable in a distributed project of this scale, which is aligned with and influenced by a number of other smaller projects. With these broad ideas in mind, the theoretical exploration led the researcher to the literature of AT and contradictions, communities e.g. CoP, colonies, collective mind, etc., learning and improvisation, which have strong resonance with the data.

In the second round of data analysis, the conceptual constructs related to the above literature were used as categories to set up code families. These codes were presented in a network view and relationships between the codes were identified. In this process, some codes were merged; some became more general or more specific. Not all code families were included in the analysis, as some were considered interesting phenomena but not directly related to the key concepts of this research. This was an intense process with many iterations until the key conceptual constructs were sufficiently refined.

In summary, the analysis reported here is the result of iterative reflections and ongoing discussions between the researcher and LCG, GridPP and EGEE members, rather than just a narrow machine derived account of the researcher's engagement with theoretical concepts. While all the quotes are taken from interview transcripts, the ideas have also been significantly reinforced by informal conversations and participant observations. Appendix D presents examples of quotations linked to key codes relevant to this study as well as examples of code families.

## 4.4 Chapter summary

This chapter outlined and provided details of the methodology used in the research. The philosophical assumptions underlying the research were described and a clarification of the choice of interpretative philosophy was provided. The philosophy was further used to justify the case study research strategy adopted in the fieldwork, leading to the inevitability of collecting qualitative data. The appropriateness of the research context for this particular type of research was discussed after which the sources from which the qualitative data were collected as well as the technique used for understanding and interpreting the data were presented.

The various sources of data collection, the unique context of collaborative distributed systems development practice under study and the constant cross-checking process and triangulation of the data have demonstrated the credibility and consistency of this research.

Having outlined the "how" of the entire research effort, the next section presents an overview of the case.

# 5 Developing the particle physics Grid

## 5.1 Introduction

This chapter provides an overview of the case. The case presented here is a combination of three sub-projects involved in the development of the PP Grid, which their nature is of relevance to this study. The engagement of the researcher with the field allowed to gain an immediate and close appreciation of the dynamics of those involved in the Grid's development and provided a clear understanding of the "key players" in the development activity.

This chapter is structured as follows: Section 5.2 presents an overview of CERN, while Section 5.3 presents a background of the case focusing on PP as a science, the LHC experiment and the need of building a Grid. This is followed by a description of the Grid's technical components in Section 5.4, after which the Grid development context (with the three identified sub-projects) is discussed in detail in Section 5.5. Section 5.6 concludes this chapter.

## 5.2 CERN: Where innovations are born

CERN, the European Organisation for Nuclear Research, based in Geneva, has been a PP research centre for over 50 years. Twenty member states are a part of this laboratory, which employs more than 6500 particle physicists from over 500 universities in 80 countries (CERN 2010). CERN staff additionally includes highly specialised engineers, technicians, computer scientists, designers and crafts people. About 3000 people are employed to construct, prepare, run, analyse and interpret the complex scientific experiments that make CERN a successful scientific organisation (ibid).

CERN is considered by the PP community to be a "remarkable home of research" and a place for fostering collaboration with a tradition of openness, working together, sharing knowledge, having freedom to improvise, having dialogue, convincing, making choices and succeeding (CERN 2004). The focus of their research has always been innovative and they, CERN people themselves, have indeed promoted that the Web, was "born" there. Having the freedom to improvise, a young scientist named Tim Berners-Lee working at CERN, wrote a proposal for an information management system based on the Internet (Tim-Berners 1996) which led to the World Wide Web, which as they argue has changed the world (CERN 2010).

As it is claimed, many creative and imaginative leaps and jumps weave their ways through the story of CERN to make it what it is today. Some of the world's largest and most complex scientific instruments have been built and used there to study the laws of nature. The LHC is their newest innovation that will embark on a new era of scientific discovery.

## 5.3 The birth of a collider

On October 2008 the most "powerful particle accelerator" went online in the world's largest PP laboratory, CERN, injecting the first circulation of accelerated particles (though quickly stopped due to magnet failures). The LHC has taken 9 years to be constructed and it is argued to be the "greatest achievement ever built to investigate fundamental physics" (Lloyd 2006).

Particle physicists describe it as the biggest microscope that will peer into the physics of the shortest distances and the highest energies ever probed (Figure 5.1). The LHC is being installed in a tunnel 27km in perimeter, buried 50-150m below ground (Figure 5.2).

Figure 5.1 The LHC accelerator (LHC accelerator 2011)

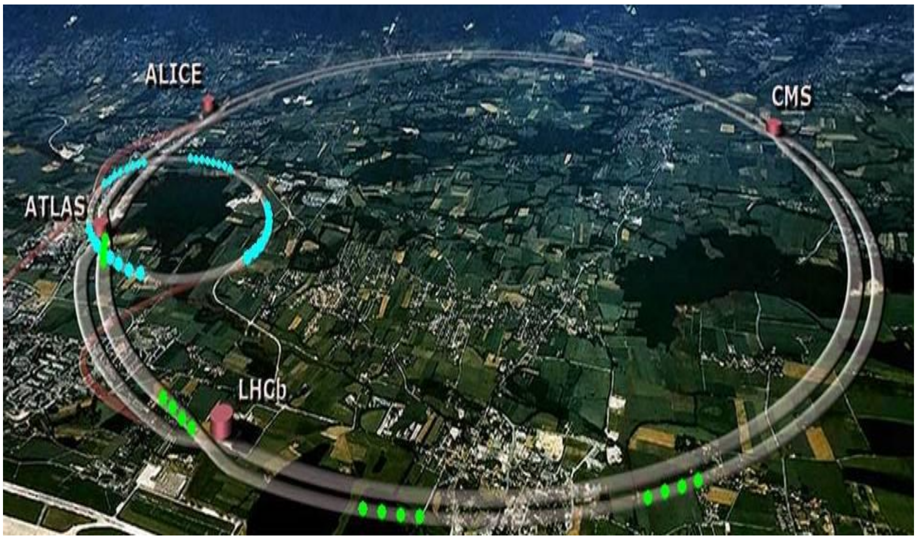


Figure 5.2 The LHC underground ring (LHC ring 2011)

For particle physicists, the LHC provides an opportunity to explore what happens when two elementary particles smash together with a combined energy of around a trillion electron volts (Doyle 2005). They expect significant new physics to occur at these energies, such as the Higgs particle, which they believe is responsible for imbuing other particles with mass and is seen as the particle that constitutes dark matter that makes up most of the material in the universe (ibid). While they argue they know that the universe started with a Big Bang, they do not fully understand how or why it developed the way it did. The LHC, therefore, is argued to enable them to explore "how matter behaved a tiny fraction of a second after the Big Bang" (Colling 2002).

When the LHC fully begins its operations, they seek for more than 600 million interactions to be produced every second (Doyle 2005). Four huge detectors being developed by the experiments already started tracking and measuring the thousands of particles created by each collision occurring at their centres (Collins 2008). The four experiments ATLAS, ALICE, CMC and LHCb observe the collisions so that physicists can explore new territory in matter, energy, space and time. The experimental detectors at the LHC are argued to generate more than 12 million gigabytes of data each year and require more than 100 000 of today's fastest PC processors to analyse it all (Britton, Cass et al. 2009). A traditional approach would be to centralise this capacity at one location near the experiments, however, in the case of the LHC, a novel globally distributed model for data and storage analysis is needed. Firstly, the significant costs of maintaining and upgrading the necessary resources for such a computing challenge are more easily handled in a distributed environment and secondly, in a distributed system, there are no single points of failure (Pearce 2004). Multiple copies of data and automatic reassigning of computational tasks to available resources ensures load balancing of resources and facilitates access to the data for all the scientists involved, independent of geographical location (Lloyd 2003).

Particle physicists have chosen Grid technology to meet the LHC computing challenge (Colling 2002; Lloyd 2006). The Grid system must allow sharing of data between thousands of scientists with multiple interests; link major and minor computer centres across the globe; ensure all data are accessible globally at all times; grow rapidly, yet remain reliable for more than a decade; cope with different management policies at different centres; and ensure data security (Britton, Burke et al. 2006a). The physicists' analysis will thus take place on the Grid network, comprising hundreds of thousands of PCs at institutes around the world, all connected to a hub of 12 major centres, on three continents that are in turn linked to CERN by dedicated optical cables. The goal of the PP community is therefore to integrate the computing resources of the several hundred participating institutes into this worldwide computational LCG.

## 5.4 Developing Grid computers

The term "Grid" was inspired by the "electrical power grid" which provides electricity via a standard plug-and-socket interface throughout an entire country, stemming from the vision that plugging into a computing grid could be as simple as plugging into an electrical grid. The "power stations" of a Grid are clusters, or farms, of computers and the "power lines" are the fibre optics of the Internet (Lloyd 2003). It is important for this study to understand the technical elements comprising the Grid and Grid middleware in particular. This section provides such information.

The Grid is divided into four layers: the lowest level is known as the "network". Just like the World Wide Web and email, Grids rely on fast Internet connections to link the computing resources of many different computers. These underlying networks allow communication between individual computing and storage elements of the

Grid. Moving up is the "resource" layer, made up of computers, storage systems, databanks and related services all connected to the network. A "middleware" layer provides the tools that allow all the resources to take part in the Grid, automates scheduling of jobs, allocates them to different computers and enables various elements of a Grid to cooperate. Finally the "application" layer allows users to interact with a top-level user interface or grid portal, from which they can submit jobs for the Grid to process (GridPP 2006b).

Although the Grid depends on underlying hardware computers, dedicated fibre and servers and communication networks similar to the Internet – it is novel software that enables the user to access these computers distributed in a Grid-like manner. This software is called "middleware", because it is conceptually between the operating systems software of the computer and the applications software that solves a particular problem and it is considered to be the key to a successful Grid (Van Der Aa, Burke et al. 2006). The Grid infrastructure for the LHC is generally built from an aggregation of many different hardware platforms, uses off-the-shelf PC architectures, such as blades and slices, runs different operating systems, and is managed by different administrations (Pearce and Venters 2011). At the same time the Grid must appear to the user application with a uniform interface from the point of view of submitting and retrieving work, such that the user does not have to have a specific knowledge of, or relation to, the resources (GridPP 2006b). This is different from the Internet where the users need to know the resources and the applications specifically and so address them through the URL that then refers to specific machines – e.g. http://www.bbc.co.uk refers to the http server application on the bbc.co.uk server. This is the task of the middleware to organise and integrate the disparate computational resources of the Grid into a coherent whole (ibid). The middleware acts like the operating system of the Grid and, therefore, allows users to access its resources without searching for or addressing them manually (Colling 2002). The middleware acts as a "gatekeeper" and "matchmaker" for the Grid, it monitors the Grid, it decides where to send computing jobs, and it manages users, data and storage (ibid). To the user, the Grid, therefore, looks like a single very large distributed system. Security is paramount in such a system and middleware provides authentication by a users' single public-key digital certificate that acts like a passport

(Kelsey 2010). Users can also have different levels of authorisation, which is administered through membership of VOs, like having visas in a passport (ibid).

The key components of the Grid middleware are: (1) the workload management, which through the Job Description Language helps describe the jobs in a way that the Grid middleware can understand; (2) the information and monitoring component such as the relational grid monitoring architecture (RGMA), which transparently combines information from multiple sources; (3) the security component which uses digital certificates issued by a certificate authority, which checks the identity of users and provides them with electronic certificates that can authenticate them to the Grid; (4) the storage element, which provides a uniform interface to storage systems in general, and to mass storage systems in particular; (5) the data management, component which supports a set of integrated data management services, such as permission to access relational databases from the Grid; (6) the fabric management component which sets up correctly a collection of nodes consisting of the site's computing fabric, in order to work optimally with each other, with application software and with other nodes in the Grid and finally (7) the networking monitoring activities which examine the network connections between sites (administrative domains), and publish this information into the Grid information system (GridPP 2006a). Figure 5.3 below presents the Grid's middleware components.

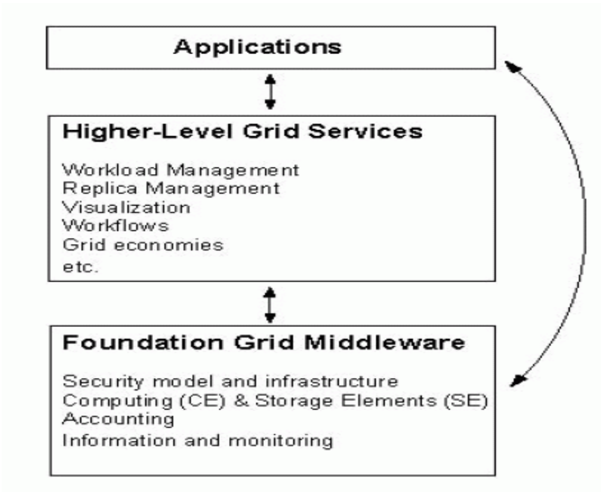


Figure 5.3 Grid middleware components (GridPP 2006a)

## 5.5 Building the LCG in context

Discovering new fundamental particles and analysing their properties with the LHC accelerator is possible only through statistical analysis of the massive amounts of data gathered by the LHC detectors, ATLAS, ALICE, CMS, LHCb, and detailed comparison with computer intensive theoretical Monte-Carlo simulations. The mission of the PP community therefore is to provide the tools and computing services to help the research physicists in their work; to build and maintain this LCG infrastructure for the entire HEP community that will use the LHC. The ***LCG project*** was built around this effort in an attempt to bring physicists from all around the world together to develop, deploy and maintain this Grid. Around 100 000 processors at 140 institutes have been linked up in 33 countries within this Grid (Britton, Cass et al. 2009).

The LCG is organised into tiers as presented in Figure 5.4. Tier-0 is at CERN and is a massively parallel computer network composed of 100 000 of today's fastest CPUs that stores and manages the raw data from the experiments (Anderson 2008). The data passed to Tier-0 by the four experiments' data acquisition systems are archived on magnetic tape. Tier-0 distributes the data over dedicated 10Gb/sec fibre-optic lines to 12 Tier-1 sites, which are IT centres located at CERN and 11 other major institutions across North America, Asia and Europe (Collins 2008). The unprocessed data thus exists in two copies, one at CERN and one divided up around the word to the Tier-1 centres. The full LCG also has around 120 Tier-2 centres, each consisting of several collaborating computing facilities which can store sufficient data and provide adequate computing power for specific analysis tasks to the entire Grid. Tier-2s are smaller computing centres at universities and research institutes. Tier-2 is where scientists actually access the data and perform their analysis (Anderson 2008). Tier-1s interact with and make data available not just to Tier-2s, but also to Tier-3s' computing resources at individual institutions.

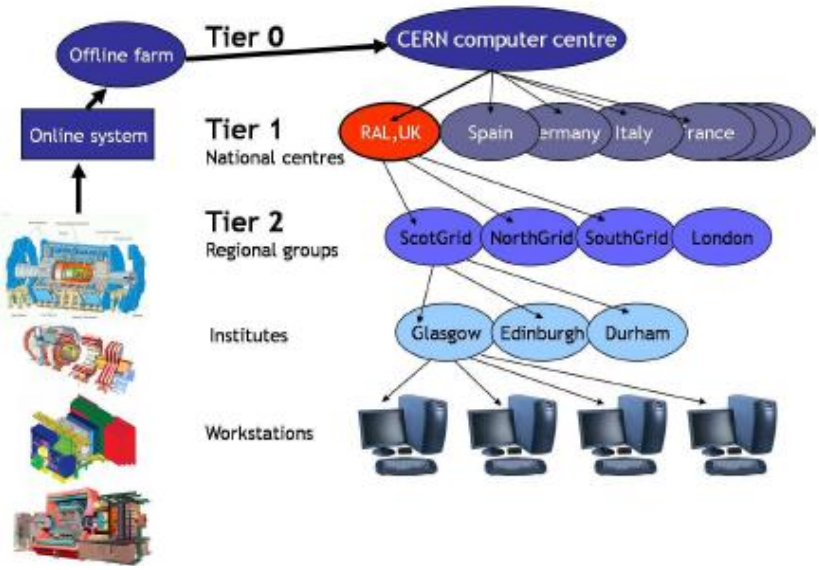


Figure 5.4 LCG's tier architecture (Britton, Cass et al. 2009)

Figure 5.5 provides a sense of the scale of the computing being developed at CERN.

.

Figure 5.5 CERN's computing hall

Grid computing for the LHC poses a number of challenges that particle physicists need to deal with, including deciding where computing jobs will run; who should have access to the data and computing resources; how access will be controlled; ensuring sufficient levels of network bandwidth; deploying software and updates across many different locations; managing data over long periods of time; and providing fast and secure access to that data. In addition, software for analysis needs to be developed that can use the Grid's resources, and interfaces should be built to allow particle physicists to submit their jobs to the Grid (Doyle 2005).

Building the LCG is a highly distributed, complex and poorly defined systems development task. Cutting edge technology and tools are used, new standards reflecting security issues etc. are being negotiated and middleware together with other supporting software are being developed collaboratively by physicists around the world. Particle physicists have a long tradition of such large-scale global collaborations and working on a distributed basis is just a part of their everyday routine (Knorr-Cetina 1999). Indeed, building this large-scale, by nature distributed Grid, demands global development. Firstly, funding needs to be taken by different sources and secondly an enormous amount of manpower and resources are needed for the different Grid elements to be developed. These, therefore, dictate that Grid elements be globally distributed rather than co-located at CERN.

The systems development activity for the Grid is organised into a number of projects, some of which extend beyond the physics community. The fact that funding is so difficult to acquire and sometimes it is politics rather than technology which may inhibit the success of such Grid initiatives (Kyriakidou and Venters 2007), means that other people beyond physics need to be involved in this large-scale development, in order to ensure transferability and usability of such a Grid in other disciplines. Other operational Grid organisations providing resources for the Grid are the EGEE, the Open science Grid etc. as well as the country-focused PP projects such as the GridPP (UK), the INFNGrid (Italy) etc.

Within this messy political context there are three significant projects of specific relevance to this study. Participants in these three projects include mostly particle physicists, both computing experts and users; however, a number of computer scientists and people from other advanced sciences are also active members in the development, deployment and user support.

*LCG project:* The major computing resources for LHC data analysis are provided by the WLCG collaboration, comprising representatives of the LHC experiments, the CERN accelerator laboratory, and all the Tier-1 and Tier-2 computer centres. The

LCG project is a PP project and is mostly based at CERN. **The LCG project is this study's point of reference** as it develops middleware components and manages the deployment and operation of the distributed computing services for the LHC on behalf of the worldwide PP collaboration (Coles 2005).

*EGEE project:* EGEE is an EU-funded project, which aims to establish a Grid computing infrastructure for e-Science, and is thus required to jointly produce middleware with the LCG project. Particle physicists heavily influence EGEE since they are the primary users of its output and much of this project is based at CERN. This research is concerned with only the part of EGEE that undertakes joint activities with LCG.

*GridPP project:* Finally, the GridPP project is one of the many country-focused PP funded projects and it aims to develop Grid middleware components and applications as the UK's contribution to LCG and in part EGEE. GridPP is here seen as a subset of LCG.

In order to avoid the complexity and messiness of getting into these projects at such a level, the Grid development community (GDC) is here defined as an amalgam of the LCG, GridPP and EGEE projects. The GDC is thus defined as those people, both particle physicists (employed as developers and users involved in the LHC experiments "acting" as developers) and computer scientists, members of LCG, EGEE and GridPP engaged in the Grid's development, deployment and user support. However, within this GDC the PP culture is dominant, strongly influencing development work (as it will be demonstrated in the findings and analysis sections); therefore the GDC reflects the organisational and managerial structures of the LCG project (and in a sense GridPP) rather than that of the EGEE. The findings section is thus geared towards describing the organisational and managerial structures of the LCG (and GridPP).

Figure 5.6 represents the GDC.

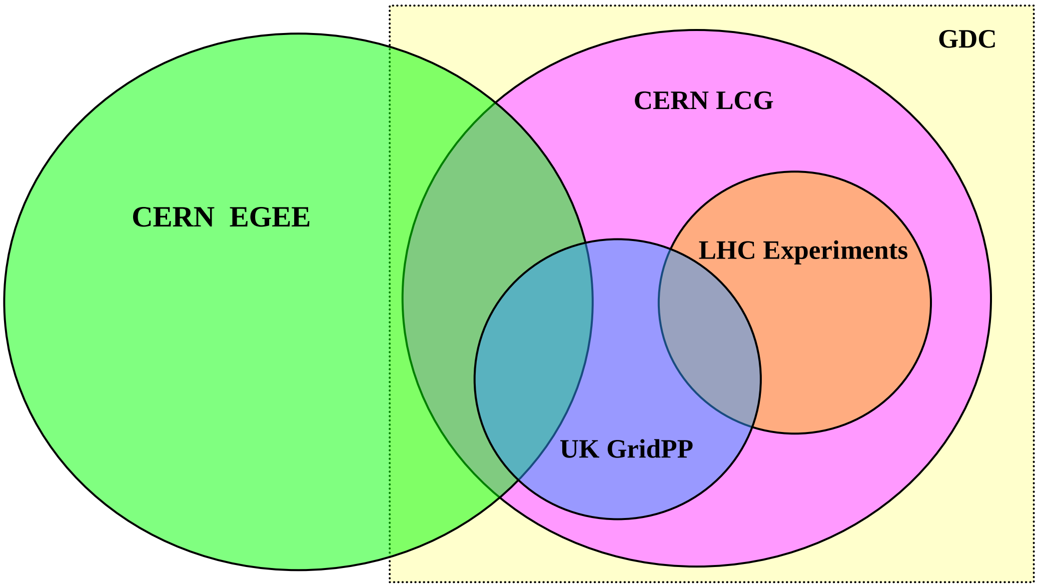


Figure 5.6 GDC: Particle physicists and computer scientists, members of LCG, EGEE and GridPP, involved in the development of the Grid

The main focus of this research is the development of Grid middleware and applications to run on top of the middleware, the deployment of middleware components and user support, all actions performed by the wider GDC. Figure 5.7 below presents this study's focus.

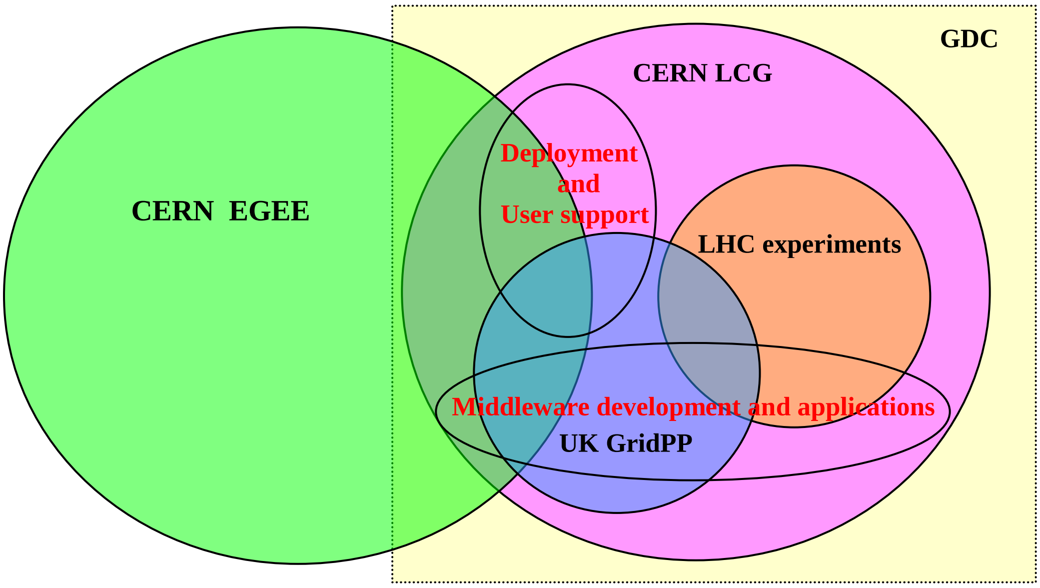


Figure 5.7 Research focus - Grid middleware and applications development, deployment and user support of the GDC

LCG presents a significant systems development challenge in technical, organisational, political and human terms, which is beyond the usual software-centric view of the development activity. The scale, complexity, need for innovation and diffuse resource base appear to defy the plan-based approaches with which the mainstream ISD literature has been preoccupied (Fitzgerald 2000). The PP community drawing on their laboratory culture and an experimental tradition that fuses developers and users has approached this task through international collective structures with a shared commitment of doing 'new physics' rather than extensive lines of authority or legal obligations. In the Grid development work itself, limited use of methodology is systematically employed apart from some *post hoc* rationalisation and documentation to satisfy funding requirements.

Various problems and challenges both of social and technical nature were faced during this effort. CERN has been keeping a relatively low profile after the LHC failure in 2008 and on the latest LHC start-up schedules for mid November 2009. The LHC had been inert for over 1 year after a magnet failure – costing almost 40 million francs to repair – which crippled operations and the Grid just weeks into the initial stages of the experiment (Novosti 2009). The Grid had to pass a series of new strict and rigorous tests that mimicked the enormous load it would take when the LHC would restart. After months of preparations and 2 intensive weeks of continual operation, the Grid demonstrated that it was ready to support the massive growth in LHC users once data taking would begin (GridPP 2009). Despite the problems, particle physicists argue that they have managed to overcome the difficulties and are now ready to enjoy "doing" new physics.

The next section provides further details on the three main projects involved in Grid's development.

***The LCG project***

The LCG project was initiated in 2002, 1 year after the initiation of the European data Grid (EDG) project as the international effort towards building the Grid for the LHC. The EDG project, funded by the EU, was founded as the flagship European project to develop a prototype Grid service, that could be used by several distributed communities (Coles 2005). The project was the first involvement of the PP community with Grids and was successfully completed in March 2004 (GridPP 2006b). After the EDG project's completion, the LCG project took the final results, such as the already existing middleware stack, and started developing them further to make them more robust, more stable and scale them up to fit the community's requirements. This effort was however merged with EGEE's effort to develop a Grid for all sciences in 2006.

The goals of the LCG project include: developing components to support the physics application software, such as tools for storing, managing and accessing physics data; interfacing batch and interactive analysis systems to a Grid environment; developing and deploying computing services based on a distributed Grid model using resources from more than a hundred computing centres around the globe; managing users and their rights in an international, heterogeneous and non-centralised Grid environment; collaborating with research network organisations to ensure high bandwidth data paths between the major LCG centres and finally coordinating the program of tests and pilot services for commissioning the LCG service (CERN-newsletter 2007).

PP collaborative work practices are not typical of all collaborations and have been described by Chompalov, Genuth et al. (2002) as ***"exceptional"***. LCG's constitution reflects these work practices and is thus based on collaboration where decisions are made based on a democratic and consensual basis with minimal levels of internal authority (Traweek 1988; Shrum, Genuth et al. 2007). The organisational structure is defined through a shared understanding and decisions are implemented and tasks are approved through the influence and persuasion rather than by strict leadership or by imposing. Although there is a loose management structure, technical decisions are largely made bottom-up, with respect to the technical knowledge at the bottom level. Leadership in the project therefore serves more as a "spokesperson" or a coordinator.

The management structure of the LCG project is best described as a network and is presented in Figure 5.8.

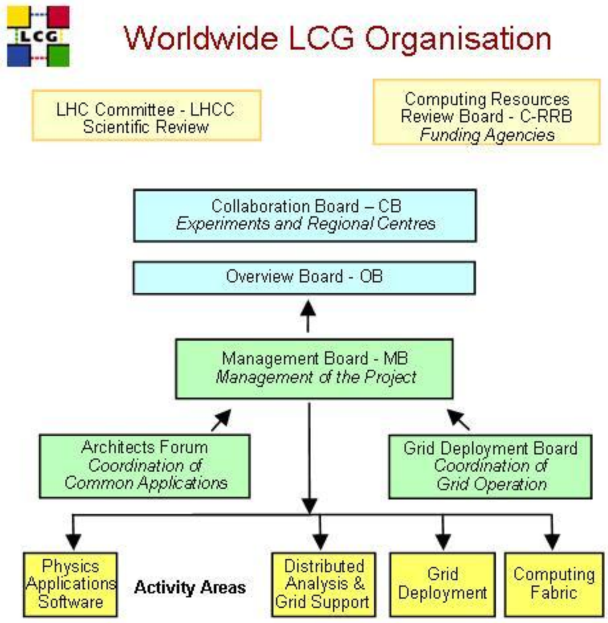


Figure 5.8 Management structure of LCG (LCG website)

The LCG cannot easily be described through an organisational chart as this underestimates its virtual, overlapping and interconnected nature. However, broadly, at the heart of this collaboration is the management board, coordinating the work. The management board provides quarterly reports to the collaboration board's overview committee. The collaboration board represents all the participating institutes. These institutes enter the collaboration bound by a "memorandum of understanding (MOU)", which serves more as a "gentlemen's" agreement rather than a contract and hence there is no authority hierarchy between the LCG and the other institutes. The MOU specifies the amount of resources and the level of service or support each site is committed to provide as well as deadlines that people have to conform with. The project's members appreciate the MOU.

Since LCG overlaps with other projects and organisations, the management board, in addition to the project leader, deputy project leader, and the project manager, consists of representatives from a number of internal and external committees, boards and functions, such as representatives from the EGEE project, GridPP project, other country-focused PP projects involved, funding bodies and other UK e-Science projects. The deployment board consists of representatives from the different countries/projects involved in the LCG and monitors the general deployment activity which is undertaken by the LCG deployment group while the architects forum represents the experiments which are participating directly in the planning, management and technical direction of the application area project activities. The LCG deployment group consists of a number of key technical experts from the LCG, GridPP and the other country-focused PP projects, including the regional network coordinators, technical experts of certain components of the Grid (e.g. storage management), and some representatives from the LHC experiments. The "leader" of the deployment group is responsible for the deployment of a production quality Grid and he or she sits on the management and deployment boards. Management roles from within the project are mainly taken up by particle physicists from organisations in the collaboration. Many people are involved in multiple boards of multiple projects and therefore each member can have more than one role in one of the projects or in different projects. Key members of the project are constantly travelling between these boards' meetings.

The systems development activities undertaken by the LCG project are varied. These involve, the development of middleware components which is a joint activity undertaken by LCG and EGEE (regional Grid structures such as GridPP and INFNGrid also develop middleware for LCG), installation and maintenance of Grid hardware, development of physics applications for job submission to run on top of the Grid middleware, testing and certification of applications, ensuring patches have been installed, user support in an attempt to identify required changes and respond to queries, etc. The project faces uncertainties, such as funding, human resources, external and internal technological progress e.g. the development and deployment of middleware provided by EGEE is immature and in places incompatible and hardware prices change. These, together with other environmental factors, "force" a more pragmatic and improvisational approach to development. Their improvisation however, not only shows the creativity and flexibility in the way they work, but also represents the dynamic nature of their practice.

***The GridPP project***

The UK's contribution to LCG is GridPP, a collaboration of around 230 people based at 20 UK universities, RAL in Oxford, and CERN (Britton, Cass et al. 2009). The GridPP project, although is the most privileged in this study, it is just one of the many country-focused projects providing resources and contributing to LCG. GridPP started in 2001 and has been involved in developing applications and middleware, deployment and support as well as in providing technical infrastructure, storage and processing units. GridPP also has been a major investor in the EDG project. For GridPP, RAL is the UK's Tier-1 centre, with four Tier-2 centres: London, ScotGrid, NorthGrid and SouthGrid, each coordinating a number of institutes in their region.

GridPP is a distributed PP collaboration that mostly consists of particle physicists, developing and using the Grid. Others within the collaboration include computer scientists, engineers and people from other advanced sciences (GridPP 2006b). Like LCG, the PP culture, tradition and style of development are dominant and provide the vision and direction to the project. GridPP reflects the practices, project management and structures of LCG, themselves reflective of PP collaboration practices. GridPP is interlinked with both LCG and EGEE. GridPP develop key middleware components as part of the LCG and EGEE projects, and GridPP also makes a contribution to the LCG deployment and operations programme. The main goal of GridPP is to provide a Grid for use by particle physicists in the UK. The experiment collaborations to which physicists belong are typically worldwide enterprises, often involving tens of countries, hundreds of institutions and up to several thousand physicists. The global nature and scale of these collaborations, therefore, means that GridPP must ensure that the UK Grid is fully compatible with their partners (Britton 2003).

The GridPP project, like the LCG project, has various unusual characteristics that pose a number of management challenges. The complex nature of the work means that it is hard to define many details of the work very far in advance. This problem occurs because GridPP is a small part of this much larger international context with which it is necessary to maintain alignment. Finally, as an academic-based project distributed across 20 institutions, there are significant issues concerning line management and culture (Britton and Lloyd 2004).

***The EGEE project***

The EGEE project was launched in 2004 with the aim of providing researchers in academia and industry with access to a production level Grid infrastructure, independent of their geographic location (Coles 2005). The EGEE project includes

40 countries and builds upon the EDG middleware stacks but introduces the production operations facilities missing from the EDG project. At the time the EGEE project started, the LCG project was already working on a robust Grid for the PP community. Doing the same development in parallel, meant duplication of effort and a waste of resources, therefore at the end of EGEE phase 1 the two projects' efforts were merged. By that time, EGEE had produced software with useful features but with low production quality, while LCG's software was based on an older technology but had production prospects. It was therefore decided that the projects should join forces and create a robust, scalable Grid that would somehow fulfil the requirements of both projects (Erwin and Jones 2008).

EGEE is closely collaborating with the LCG project and GridPP for the development and deployment of the Grid middleware. There is lot of overlap between the EGEE and LCG projects. LCG is the PP component that has links into EGEE, and also into PP experiments. Various people who work for LCG and GridPP also work and provide resources to EGEE. Both LCG and EGEE developers develop middleware components. The EGEE project consists of computer scientists, other scientists, managerial people, etc., but mostly particle physicists. Particle physicists within EGEE are involved in the development work but are also the largest group of users and the most expert computing group of this Grid infrastructure. Historically, particle physicists provide most of the resources on the EGEE, because they use most of the resources. EGEE have therefore found themselves on many occasions struggling to distinguish between their goals and those of the LCG project and the PP community.

EGEE is an EU-funded formal project with clear objectives, management structures, timetables and deadlines for deliverables and industry, so-called "best-practice" techniques are aspired to be used during the development (EGEE 2010). The EGEE project is organised into 10 "activities", which come under three main areas: the networking activities which are responsible for the management and coordination of all the communication aspects of the project, the service activities which are in charge for the support, operation and management of the Grid as well as the provision of network resources and finally the research activities, which concentrate on Grid research and development of middleware.

## 5.6 Chapter summary

This chapter provided a description of the case under study. An introduction to CERN and its major achievements was discussed, followed by the LHC experiment's need for the development of a Grid. This chapter also portrayed the research's focus by explaining why a number of projects (LCG, GridPP and EGEE) informed the development context as understood by the researcher.

The projects were presented in detail discussing their structure, objectives and the general culture of people involved with more focus on the LCG that is the point of reference to this study. The study presented in this chapter sets the ground for a thorough discussion of the findings. This is the focus of the following chapter.

# 6 Findings

## 6.1 Introduction

This chapter provides a detailed presentation of the empirical work, unpacking the dynamic, flexible and improvisational nature of the GDC's collaborative systems development practices when developing the Grid. In particular, focus is given on the social and collaborative dynamics that drive the Grid's development and how these have shaped the systems development practice that followed. The key questions driving the rationale of this chapter are: What is the nature of systems development for such a Grid project? What kinds of systems development practices are employed? How is the development, deployment and user support being done?

In order to unpack the dynamic development of the Grid and how such a collaborative distributed effort is organised, managed and coordinated there is a need to structure the findings in the following way. The first part of the findings section starts by describing the need of the PP community to develop a Grid and the alliances created in order to achieve this. This is then followed by a thorough description of how this collaborative development effort is managed, pointing to the interesting and distinctive characteristics of the PP members involved in the GDC such as the shared goals driving the work, freedom, trust, consensus and democratic decision-making. Finally, the last part of this chapter describes the development, deployment and user support with a particular focus on the practices, methods and tools employed.

In exploring these issues, the development context, as well as the different groups of people involved, are firstly described in Section 6.2. How this whole effort is coordinated and organised then follows in Section 6.3 with particular focus on the "management" structures and tools for planning, the division of labour, the communication channels and the decision-making process. Section 6.4 discusses about competition and funding pressures that present a challenge to the development of the Grid after which dealing with systems development is explored; details on how the development is performed are provided, by discussing the different practices, tools and solutions employed (Section 6.5). Section 6.6 summarises the chapter.

## 6.2 Forming virtual alliances to build the Grid

***In need of a Grid...***

The LHC is considered by particle physicists members of the GDC (from here on defined as PPGDC) to be *"the biggest and the most challenging opportunity in the history of science to discover how the universe behaved a tiny fraction of a second after the Big Bang"*. The development and deployment of a Grid is an essential and compulsory part of this effort for storing, distributing and analysing the data of the LHC: "*It would be very difficult and would have taken us longer if we didn't have the Grid...Even if you have the best detectors with the best security in the world, if you don't have the software to analyse your data and to extract the correct information from the data, then all your money is wasted".* PPGDC see no alternative but the Grid in order to be able to perform any serious kind of analysis: *"It has to do with scalability. It is impossible to build a traditional batch system of the size that we need for LHC. That is why the grid is here".* Building this Grid is also seen as crucial because of the nature of the PP work. Their experimental physics work requires large-scale distributed collaborations to be assembled where thousands of physicists work together to produce scientific results; therefore the Grid provides an opportunity to access all data equally. This collaborative nature of PP as a community is indeed somewhat unique; the only way to produce results is by collaborating, compared to other communities whose work is more individualistic and independent. Hence, building the Grid, has proven to be an attractive option as it can both facilitate the analysis of their scientific work as well as support their distributed collaboration.

Although recognising the difficulties behind building the Grid, their extensive experience in distributed computing gives them sufficient faith and a strong belief in their ability to overcome technical obstacles in order to construct the necessary Grid infrastructure.

***The Grid as an evolution of a long computing tradition***

Mobilising to build the Grid came quite naturally for PPGDC, with some even stating *"we would have done it anyway"*. The long computing tradition within the wider PP community means that until recently PP has been self-sufficient in developing its computing resources relative to other sciences or industry. This IT expertise is mostly related to the insufficiency or non-existence of commercial off-the-shelf products to facilitate their work. It is argued that *"no-one is going to do the work for them"* and therefore they have on numerous occasions claimed that *"you either don't do the PP or you learn how to be a computer expert. We breed a lot of geeks, and I mean it in a nice way, who really like doing computing. And because they are so tuned in with PP and see it as a worthwhile goal, they do this computing not as a job to just do computing, but because they want to see the PP done".* PP has always been at the forefront of the development of scientific computing with Tim Berners-Lee's invention of the World Wide Web at CERN regularly used as an example. Building the Grid is just seen as building upon their previous knowledge: *"There has always been a long history of computing within the community…we always had lots of computer experts, we always developed software for the experiments like tracking codes, reconstruction codes and analysis codes, etc...So it was only natural to build this"*.

However, computing is regarded as being of secondary importance to the PP community. It is perceived as a tool that allows them to do the physics and supports their analysis. Their interest lies in *"doing new physics"* and how the technology can help deliver their scientific goals. PPGDC with IT expertise, therefore, approach Grid computing with a focus on achieving their community's specific aims rather than being concerned with IT per se. An experimental physicist indeed argued: *"I do this because I want to find the Higgs and if it would be easier to do it with punch cards I would do it with punch cards…We need computing like we need accelerators, and like we need mathematics to understand our mathematical models. And you need to have skills in all of those fields to do what we do. People really become very flexible, they learn a completely new field if they need that for their work. So in some sense, computing is just one of the things we had to learn".* The LHC delays, nevertheless, mean that even more PPGDC spend time doing computing, because *"they have no data to analyse and since they cannot really do physics they do a lot of computing".*

***Joining forces: LCG and EGEE***

It was realised early on that the development of such a global infrastructure would present a major challenge for the PP community since it would require them to form distributed alliances with a range of people and agencies across the world, including funding bodies, universities, and commercial enterprises and with projects with wider goals such as the EU-funded EGEE project. For example, the need to secure funding for the LHC's computing requirements meant that the UK PP community had to align to the goals of the UK e-Science in 2000 – 2001 in order to gain the funding which this programme provided. Developing the UK Grid, a part of the GridPP project and thus the UK's contribution to the LCG, became aligned with the delivery of the UK e-Science requirements, alongside providing the necessary new computing resources for the LHC. A senior developer commented: *"Although CPU and disk storage become cheaper each year, there was insufficient funding available at CERN to both build the LHC and meet its computing requirements. Substantially more resources, both in terms of hardware, technical support and funding, were distributed through CERN member's home states. Therefore, despite considering the attractive option of concentrating all computing resources centrally at CERN, it was found to be impractical for a number of reasons"*.

The Grid's technical elements thus demanded a distributed collaborative effort to be organised around their development: *"It was impossible to put all the computing power into one single place. Not only politically because funding agencies tend to want to see something for their money and so they want to see the computers in their country, but also technically"*. The PP community's response to this problem was to approach it in a similar way they would approach a physics problem; thus to create a large-scale computer-supported collaborative distributed community around the Grid development effort (the GDC defined in the previous chapter) with the aim to solve this global problem: *"There is now a community that exists that did not exist before, that's for sure. It will continue to exist because we have to solve this problem...I don't know how else we would do this Grid development which is very decentralised, apart from collaborating"*.

The LCG project was initiated to act on behalf of the worldwide PP collaboration on this distributed large-scale development effort. Smaller PP country-focused projects were also initiated in an attempt to contribute to the Grid's development, including GridPP, and other projects with wider goals including EGEE were involved. A senior member at CERN indeed argued that: *"The LCG was set up specifically by the LHC* *community to understand how we could make the computing environment work. It just happened that grid technology came along at the same time, and it looked like a good fit. So we said – OK, take this technology and see whether we can make it into a service. I think that got enough momentum behind it so that then EGEE was proposed initially as a way to fund the operation behind this and continue the development of the software. And so people went to the EU and said – this is what we propose to do. And the EU is funding it as a research project. It is set up as a research infrastructure. It is not pretending to be a commercial enterprise"*.

The complicated international context in which all three projects of the GDC (LCG, GridPP, EGEE) are embedded and the complex nature of the work programme leads to a project definition that is "incomplete". They do not produce, or even sub-contract, all the components needed to produce the final product; instead, they rely on developments from each other or other related projects. This interestingly reflects EGEE, which despite being a more formal and "industry-like" project it still relies on others for the development of a usable Grid for all sciences. The GDC's projects' definitions are thus "dynamic", in that there are frequent changes at all levels, and to some extent the projects are 'devolved'. As a GridPP technical coordinator explained: *"At the technical level decisions are made outside of GridPP. We are part of a worldwide group of PP and decisions have to be made on a worldwide basis and fit in with LCG and EGEE. I wouldn't say CERN decides, but the major activity tends to be centred at CERN and so whoever's coordinating a particular activity, they will generally seek common consensus in reaching decisions"*.

Middleware components and Grid applications are being developed in a distributed fashion; for example, the data management component is developed in collaboration of a number of UK and Italian developers funded by different projects and sources. This distributed setting however, poses an additional challenge to the formal management of the GDC's effort. Various novel tools and techniques need to be developed to manage these challenges.

The following section provides details on how the Grid development effort is organised by emphasising on the management structures, the division of labour, the planning process and the communication channels of the GDC. The GDC, however, reflects more the managerial practices and processes of the PP projects (LCG and GridPP) involved, rather than that of EGEE. Therefore, this part of the findings is geared towards describing the LCG's managerial structures. This in a sense reflects GridPP, which is a subset of LCG.

### 6.3 How this decentralised computing effort is organised and coordinated

Particle physicists have always seen themselves (and are seen by many) as the elite among the sciences (Chompalov, Genuth et al. 2002). Traweek (1988) even described them as "promethean heroes of the search of the truth". Particle physicists have a long tradition of being a collaborative community, with Porra (1999) describing such a type of collaboration as a colony and Knorr-Cetina (1999) describing them as egalitarian and communitarian. Their collaborative way of working is recognised as participatory "Athenian democracies" (Shrum, Genuth et al. 2007) and traces back to their history, their culture and the nature of their experiments seen as collaborations: *"Particle physicists have collaborated anyway...we run big collaborations across every nation on earth and they always work".*

Rather than searching for traditional project management structures or techniques for the Grid, PPGDC draw on the history and culture of the key leaders (all of whom are particle physicists) and structure the PP projects of the GDC (LCG and GridPP) like a PP experiment. A GridPP team leader indeed commented that: *"The original proposal is set up deliberately to make GridPP look like an experiment. This whole idea of collaboration board for instance, comes out of the idea of what an experiment does. So almost by design, it has been made to look as an experiment. The difference is the day-to-day things that might be different between an experiment and the project. The project is also very distributed and it is less easy to identify things. That is why we have a lot of reporting, meetings and things. An experiment also has to encompass lots of things, e.g. hardware, physics, as well as computing. Whereas this Grid project really is primarily computing".*

With decades of experience of running experiments, it is perhaps unsurprising that the LCG project and GridPP are set up in the structure of an experiment and largely managed in the same way. An LCG physicist indeed explained that: *"How we work* *now is not different from how we have always done experimental collaboration before doing this specific computing side of it. Now we see it as an extension of the sort of stuff we have always done. So it is like doing collaboration work in a physics group"*. There is a feeling that the work is "experimental" and thus different from other perhaps very similar infrastructure development projects (e.g. within industry). Interestingly, PPGDC believe that their established traditions and accumulated experience within physics mean that working in large-scale globally distributed collaborations is *"second nature"*. Furthermore, the development of the Grid on the basis of needing to achieve a scientific goal translates their programme of action into "getting the job done", which together with their experience and tradition provide a solid basis for the improvisation of the Grid development, something that leads them to believe that this is sufficient foundation for the LCG and GridPP projects.

### 6.3.1 Organic, flexible management structure with a strong shared vision

Although a management structure is put in place for the PP projects of the GDC and an extensive structure of management boards, committees and technical groups exists, these serve more as communication channels between clusters of expertise, than a hierarchy of authority. It is argued by LCG senior members that this structure *"has been decided consciously"* and is in some ways seen as *"an organic structure, because it is evolving"*. Indeed, although the boards have been envisaged from the start, it is agreed that *"their roles have evolved, to natural niches in the information ecosystem of the project and they do all seem to somehow manage to function and get the job done"*. Managerial roles within LCG and GridPP serve most of the time as representatives, spokespersons or coordinators and when decisions have to be made centrally at their management boards, such decisions are open to inspection by the full collaboration. Interestingly, it is claimed that "*even the management board members are not managers, they are coordinators"*.

Whilst this might sound a bit chaotic, a GridPP Tier-2 technical coordinator commented: *"When I joined the project I could not believe the number of committees and boards and forums both within GridPP and LCG and often it is exactly the same people sitting on them with different hats. It looked like a completely dysfunctional organisation when you saw its management structure. But now having worked on the project for just over 2 years, I think it does work. And I think the reason it works is* *because ultimately it is successful in gathering the inputs from the different interest groups…so somehow the worldwide organisation manages to successfully capture requirements and process them into outputs. And although we maybe all moan about spending a bit too much of our time in meetings, the process does actually seem to work".* PPGDC argue that this structure makes sense, however, there is still disquiet and concerns expressed from the computer scientists members of the GDC (from here on defined as CSGDC) who describe LCG's and GridPP's structure as one with "no teeth" which lacks the discipline of a company. They believe that the scale and the geographical spread of the GDC's projects limit the control people can have on others; something which creates confusion, delay and frustration which cannot be resolved because of the limited management control "*within LCG and even within EGEE to a certain extent"* and therefore there is a need for a lot of *"pushing and shoving and talking and a lot of meetings and frustrated emails"*.

A project manager explained that members of the LCG project (and GridPP) e.g. users, system administrators, deployers, developers, etc. are not organised in what he believes to be an industry-like way but in a way that he believes brings about truly collective forms of working. While some PPGDC may criticise "industry-like" ways of working as lacking this sort of collaborative culture, what is interesting is the lack of experience within this PP community of industry. There are indeed studies that suggest that industry is collaborative, but what appears to be different here is the amount of trust, autonomy and voluntarism. PPGDC see the LCG and GridPP projects as a collaboration and have many times referred to them in terms such as a "*kind of a federation club of smart academics who all want to do it and everyone trusts each other to be doing the best they can".* In traditional PP groups, the group leader is not seen as a leader in the conventional sense, rather they are coordinators who trust their team to perform well. It is argued that: *"It is never necessary for a group leader to say – you're not working hard enough or you are working on the wrong thing or check if you did that, as people only get appointed if they are good and the important thing is that you know you can trust people to get on and do a good job"*. It is this fundamental trust that appears to drive the PPGDC. They believe that it is this culture of trust and equality that makes people want to "*step up to the plate and do the dirty work as well as do all the glamorous work"*. The nature of the PP work requires having faith and trust in what other people have done. Interestingly, it is claimed that people in this PP community are respected for what they do and not because of their job description or status. Most importantly, there is an enormous respect to the technical knowledge at the lower level. This is one of the reasons why they describe their collaboration as a *"democratic meritocracy"*.

The same ideas are of course applied to the LCG and GridPP computing projects having a group of computer scientists and physicists working on the development. As an LCG project manager commented: *"There is no hierarchy of you know I am the boss, I will tell my guys you will develop this software. It only works if they want to develop the software because they understand why it is important and they are interested in it; that comes from the PP tradition"*. Individuals or groups in the projects will try to solve a particular problem or develop an application, not because their manager told them to, but because they feel that it is useful for the whole project: *"This environment is based on, if you want, charismatic leadership and people doing things relatively independently but also having the freedom to do them, and not having to report every 2 minutes on what they are doing"*. They are generally given freedom to carry out their work, usually without clear instructions or strict supervision and this is because, in their view, the PP community involves people with commitment, intelligence and self-motivation.

Interestingly, when asked "How do you know what needs to be done in your job?" most PPGDC indicated that they respond to emergent issues or they just look around and find problems to solve without necessarily being relevant to their job description. People volunteer to do things and shift between jobs not because they are forced by someone (as they argue), but because they want to (though political forces can obviously play a part). As a PhD experimental physicist argued: *"Last summer we had a big testing campaign and they asked for volunteers, it was an 8-hour shift, and it could be a night shift, but everybody did a bit. I did 18 shifts, 18 times 8 hours, and that was of no use to my PhD whatsoever. I learnt things there which I couldn't have learnt in any other way, about the experiments in general, and it is the only way you can learn those particular things".* This behaviour is seen by them to be an inherent and natural component of the PP practice. It has been nurtured over the years and it is based on their physics tradition. Indeed, as a Tier-2 manager indicated, this "volunteering" way of working of PPGDC is driven from their love of doing physics and their need to feel that they have contributed to this "sacred cause": *"People often quote that the reason why we are so successful is because we've got lots of highly self-motivated people who will get on and do the work in spite of the management. We don't work for money of course because PP doesn't have a lot of money. We work because of our passion to do science. So we strive to deliver the best result. To collaborate in the best way because we serve the same ideas and the same passion".*

It is therefore unsurprising that the way this Grid computing effort is organised is indeed by encouraging people to work in a way where a shared vision is built at all levels. This shared vision, the shared milestones and the shared deliverables are, in their view, what bring the collaboration together and enable the different people from different parts of the projects to work together so that they feel that it is their job to make this work. The common goal of doing physics appears to provide an important source of motivation and a strong sense of direction, urgency and progress as well as binds efforts and bridges differences. As a senior LCG member argued, because of this shared vision, you can get physicists to do a certain amount of *"tedious stuff"*. An LCG physicist/storage developer interestingly commented that: *"We are one community, we have one goal and we are all working towards that. So I've done nightshifts and things like that because they have to be done and there's a sort of tax on everybody that they're all prepared to pay to be part of the collaboration and get access to the interesting stuff".*

There is a strong belief that PPGDC have a unique shared vision and the ones interviewed constantly discussed about the shared goal that drives them. They indeed believe strongly that this shared vision leads them all to collaborate in such a way. This is, however, the voice of those selected to stay in this community. One should be mindful that the high turnover of staff may indicate that not everyone shares their views. People do leave and it is not possible to understand why they do. There are also a lot of junior PPGDC that are looking for jobs outside the PP community, because of poor work conditions, lack of job security, etc. Furthermore, the CSGDC appear less keen on this shared vision of doing "new physics" and thus as some of them argue, they *"do not feel as though they are part of a community where they should contribute"*, something which comes as a surprise to most PPGDC realising that other people are not *"in the same habit of collaborating"* as them. Those who remain in the community and are successful within it, however, are those who are inspired by this shared "sacred cause".

This shared vision, therefore, which is not essentially to build a Grid, but more importantly to do new physics and try to understand the origins of the universe is argued by most PPGDC to facilitate a commitment to enable and support the collaboration and foster a strong community bond. In their view, this collaborative attitude that was nurtured over the years, it is sustained by building a strong sense of belonging in the community, which they believe makes people align themselves with LCG and its objectives.

With members of the wider GDC being so dispersed, it is important to develop social and emotional bonds among individuals for this effort to function collectively. A middleware developer stated: *"We have to work well together as a team in order to be successful".* This sense of belonging is facilitated by the shared culture in the PP projects of the GDC, which emerges from the physics background of most participants linked to the memories of previous successful and innovative experiments; to a history of cutting-edge computing and to the tradition and culture of strong commitment with a long-term vision and pragmatic problem-solving. Interestingly, there are a number of people within the PP community who have been working together for more than 20 or 30 years, demonstrating the strong continuity of this collaborative working: *"There is a strong continuity of collaboration with the same people and that helps a lot. The community already existed long before LHC. Long before LCG existed, the people working in the physics community are well established. So the context of collaboration is there".*

One however could argue that this is a closed shop of senior people at the top with difficult access from below. The people working in LCG and GridPP are well established, which makes it difficult for newcomers from different disciplines to enter the collaboration. As a computer scientist member of the Grid deployment group, jokingly stated: "*I think the particle physicists like to think they could have done it all themselves without any help but, yeah...there was a joke that went round* *that the only way we got computer scientists into the project was [because] they obviously had to sneak their way in".*

Furthermore, this existing culture of collaboration does not mean that competition or politics do not exist. On the contrary, it is argued that although the PP community is collaborative in the broader sense, competition still exists. The LCG collaboration is described by some CSGDC to be "symbiotic" and it has been even linked to ecology where people are seen as predators. Indeed an EGEE coordinator commented that: *"You have to realise that this is an ecology. It is an ecology where there is a certain element of competition and getting a solution that does the job, even if it is not beautifully written, is very important, because you could be designing the best system ever, which unless it gets adopted is a waste of effort. So you need to get users using your system quite early on and that is closer to the way particle physicists do things than the way computer scientists do things".* While PPGDC work as a collective they still compete and remain as individuals holding different interests. Fighting for resources and funding are just some of the reasons to compete. As an LCG computer scientist stated: "*As experiments get larger you quite often have competing technical solutions, which, if you manage to get collaboration working well then they can merge to become a common solution. But sometimes people want a shoot-out where one lives and the other dies…and then it is more a war between teams so that they get money".*

PPGDC may be ambitious and competitive but they nevertheless need each other and CSGDC and realise that they cannot survive without the other. Although their research papers are published with hundreds of authors, they still argue that they know who contributed what. Like players in any sports team they argue that they balance their own performance with the need of the team expressed as shared goals. As one interviewee put it: *"We are trying to get to the data that comes out of the LHC and there are times when we know that we will compromise our own parochial little gains to reach that higher goal. This high level common goal makes it actually easy for us to do this thing [to collaborate, to work together]".* This demands fewer traditional managerial structures, but it also demands a great deal of collaborative skills such as emotional communication, building consensus, fostering a sense of belonging, building of trust and "just hanging out".

PPGDC draw upon their experience in large experiments, their competence in computing, their ability to work in large distributed groups with people from different disciplines and cultures and their tradition of breaking down a complex task into smaller pieces in order to investigate, experiment, and move forward by trial-and-error. They have restructured the task of developing the Grid to be one that they are largely familiar with – a distributed experimental collaboration – which can be seen as just another task that they have to complete in order to achieve their common goal – doing new physics. This way of working is learned over the years and has deep roots in their *"history and culture*" as well as resides in their *"desire to sort of jointly achieve things".* It is in this tradition that they find valuable lessons that help them organise the Grid development effort and make it work: *"The way we work is probably the thing that makes this LCG project work and this relates to our tradition in working in such large collaborations. You cannot get this number of talented people all working at one site. We have ways of working with that which have developed over the last 30 years – for example the telephone meetings, the several per year collaboration meetings, where people try to get together…And I guess those have evolved and this Grid development is also being done in a similar vein, because it more or less works. We have somehow learned how to organise things at the project management level and how to take the pragmatic view when faced with a problem in order to find a solution. So we have this tradition in problem-solving and the sort of pragmatic approach in project management".*

### 6.3.2 Mutation of roles

This organic and flat structure of the LCG project (and GridPP) precludes any clear division of labour. Individuals can shift between jobs and have more than one job at the same time. The priorities for the development work change rapidly so that the PPGDC and even the CSGDC find themselves having to recurrently embody different roles such as those of integrator or maintainer.

When asked about the integration of expertise between computing and physics, the answer was that certainly within the wider GDC a large number of PPGDC regard themselves as both developers and users. PPGDC are "powerful users" who will encompass the role of developer if they are not satisfied with what is provided. An experimental physicist argued: *"If there is some of the official grid technology which* *isn't working then we just bypass that and replace it with a home-grown replacement. Our primary purpose is to analyse the data, if we can do it on the grid, that's fine, but if it gets in the way I am sure we'll just chop it out and produce something else. That's the physicist approach".*

PPGDC themselves *"don't regard their roles as having fixed boundaries"*. A physicist middleware developer indeed claimed: *"We tend to do things which we are better at regardless of whose role it might actually be"*. Interestingly, the coordinators of the LCG project (and GridPP) encourage this kind of behaviour, not only because it is rooted in the community's culture, but because they *"recognise that you keep people efficient and alert by letting them do what they're interested in and what they are good at"*. Furthermore, they believe that in the PP community *"people are always really willing to learn and take on advice"* and therefore accept an importance for allowing space for creativity and innovation. The invention of the Web is regularly cited in defence of this focus on individual creativity. For example, as one senior CERN interviewee who worked closely with Tim Berners-Lee put it: *"Why was the Web invented here? Because Tim had the freedom from this hierarchy to spend a bit of time investigating something which was of interest to him, and nobody else here said – 'oh it's a waste of time, never mind'. He was working on remote procedure calls. And out of it popped the Web…One guy, sitting in his office, who had a dream".*

### 6.3.3 Strategic planning or "planning not to plan" and prepare for change?

There are difficulties in this virtual complex and messy Grid development environment, suggesting all three projects of the wider GDC to be flexible and to be able to quickly adapt to changes. For the PPGDC this complexity and uncertainty suggests that the development work should be *"more day-to-day stuff"* and leads them to believe that "vague milestones" and lots of improvisations are required. As a GridPP developer stated: *"We had set lots of milestones which at the end of the day were found useless compared to the day-to-day targets created by us. So setting milestones ended up being irrelevant because we revealed a lot of problems which just couldn't be resolved on that day"*.

Computing on a global scale means that requirements are difficult to pre-specify. In addition, collaborating in the international arena makes planning harder since all the GDC's projects must closely align with each other's goals, needs and expectations. Therefore, it is complicated for LCG, GridPP and EGEE to jointly plan the software development activity and set out very clear milestones and deliverables for the long term, resulting in the PPGDC and the CSGDC just prioritising the things that matter. For EGEE the rigorous demands of EU funding demand a creation of these milestones, but these are not believed to be taken seriously by the LGC and GridPP.

In LCG and GridPP, fluid practices that serve as a continuous response to external and internal change are observed. Grid development is not guided by clearly articulated plans or structured methods. It depends on improvisation and spontaneous actions in order to deal with problems, unexpected opportunities and changing requirements. However, while improvisation is one way of coping with complexity and uncertainty, strategic planning, although minimal, is certainly not absent. Within both LCG and GridPP, it is recognised that the ad-hoc practices of dealing with the unknown have to be supported by financial planning, project milestones and resource allocation mechanisms. Planning is recognised as crucial for providing the fundamental legitimacy of the project since reporting to the Oversight Committee on time means better chances for securing future funding and higher appreciation and recognition of leading a "successful" project. A senior LCG physicist developer indeed argued that *"management attitudes have changed towards milestones and oversight committees. It wouldn't really have mattered so much in the older days what people did, the funding probably wouldn't have depended too much. Now it's very rigorously related. You've got to justify your existence every 2 or 3 years to get increased funding. So there is that pressure all the time"*.

Extensive Gantt charts and schedules are produced but serve as a minimal structure for the projects, providing the foundation and direction necessary and *"guidance on how things are supposed to work"*, although PPGDC know very well that in reality nothing goes as planned. They claim that: *"although you try to do planning in advance, you won't have thought of everything, things change constantly so it doesn't worth putting a huge effort"*.

The LCG's and GridPP's management boards' focus is on supporting and explaining change. An LCG project management member argued that: *"We wanted to establish the fact that we had the right to change our deliverables. So we set up this project map and we set up the formality of change forms. So this was to formalise our freedom to change the project… yes, we had a set of milestones but you know, we had a mechanism to change them because we had to be responsive"*. The projects maps and schedules are constantly changing but together with the quarterly reports and Gantt charts become instruments created to achieve various goals: to demonstrate rationalised order, to obtain legitimacy for the project, to provide momentum, to cope with changes, uncertainties and new requirements and to support or legitimise their spontaneity. They also help align the different distributed innovative actions and priorities with the collective goal, although the plan may be emergent and layered out in their day-to-day sense-making and actions.

As a GridPP technical coordinator argued, the project is *"visionary…there is this vision of the Grid paradigm and this way of working, and what has happened is [that] everybody tries to catch up with that and make things work so it meets that vision, rather than the other way around".* This proactive mode of management is also combined with a reactive mode of daily troubleshooting. These are interestingly seen as flexible management tools and it has been argued by a GridPP middleware developer that *"the structure and management of GridPP with this project map defining all these milestones and everything is what binds the project together. So if you had asked me a couple of years earlier, I would have probably said that this is what I dislike. But now looking back, I can see that to have a successful project and for it to be measured, you needed all these things. So I think that actually has been done rather well and probably better than in the experiments in some ways".*

In other words, as Zheng, Venters et al. (2007) have observed there is a plan to improvise and carry the project forward by improvising pragmatic and practical solutions and by enacting processes that encourage improvisation. This ability to improvise is again seen as the result of years of experience and learning. Such improvisation represents a further process of exploration and reflection that feeds into the community's ability to improvise (ibid). As one of the technical coordinators nicely described: *"we are in a foggy valley with a goal that we are roughly marching* *towards...You need to retain enough of an idea of the general direction which represents progress, and the very specific goals which advance you. So if you like you need your head in the clouds to see the big picture, but you very much need your feet on the ground because you have to put one foot in front of the other, and day-to-day we keep putting one foot in front of the other".*

### 6.3.4 Communication channels coordinating the work

While the development environment may seem chaotic, the actual day-to-day development activity is not undirected. PP collaborations are managed and coordinated by what Knorr-Cetina (1999) refers to as "a fine grid of discourse", which channels individual knowledge into the collaboration providing it with a sort of "distributed cognition" and a stream of "collective self-knowledge", and which flows from the amazingly composite webs of communications. For LCG (and GridPP) this web of communication is horizontal and attempts to *"give people this broader vision of the project".* It includes a complex network of boards, committees, and working groups, involving individuals or groups from different layers in the collaboration, and which are regularly holding meetings, either physical or virtual. A GridPP middleware developer argued that: *"There are clearly multiple routes by which information reaches the ground here [UK]. So sometimes it comes directly from the LCG management board, other times from EGEE, sometimes through the development team meetings, or our local users, so there are lots of different channels…and that can be very disconcerting when you join the project. There are many lines of communication. It's unclear what each of them is supposed to be communicating to you. But after you've been working in the project for 6 months, say, I think people feel quite comfortable with this, and feel that it does work".* As a GridPP project leader explained: *"The size of the collaboration has increased and therefore it has to become more democratic with lots of boards, committees, processes, structures and a number of communication channels to make this whole thing work"*.

It is indeed argued that members of the wider GDC (LCG and GridPP and EGEE) maintain a general understanding of the work and the shared vision and especially related aspects to their specific roles and daily activities, through these continuous and extensive communication flows in the community. As a team leader explained:

*"You need multiple committees because there is no centre, you can't force things down on people, so you need to have different forums that requirements can be gathered, and different channels of communication for pushing things from different angles".* They commonly agree that there is a need to ensure a good communication flow at all levels: *"It is essential for people to know the overall picture and see where their work actually fits into the overall picture as well as make sure that if people have good ideas, these are communicated up".* Members of the GDC are not part of a coherent institution responsible for managing the development process; rather they are *"a chaotic set of collaborators with limited control over what each one is doing"*. There is a feeling that members of the wider GDC *"have to be actively involved, go to all the meetings and belong to different teams to know what is going on"* and in order to *"maintain focus and commitment and enough interest in delivering the service for PP".* As PPGDC argue, this is because in such a distributed project collaborators may "*miss all the coffee conversations and corridor talks"* and therefore they need to find other ways to compensate for that.

It is argued though that there is an overload of communications, meetings and emails with some PPGDC even stressing that *"at some point you just don't do anything but sit in on meetings and exchange emails and argue with everybody else"*, which they acknowledge as a waste of time. Nevertheless, they do not regard themselves as having fixed working hours. As a GridPP physicist computing coordinator said: *"We could spend more than 24 hours a day; we do nightly shifts and we even work at weekends".* While they may spend lots of time in attending meetings, they still find the time to do their work.

There is a common agreement that although *"face-to-face communication is pretty labour intensive and is something that requires travelling",* it is still "*not possible to do everything just through video and email, although that does help".* Different Grid components have to fit together and therefore it is necessary for people involved in the different GDC's projects to meet every few months to discuss various development issues that are common to the projects.

Most daily or weekly meetings for coordinating and dealing with issues around the development, deployment and user support are conducted virtually, although a number of formal face-to-face meetings (such as the WLCG or GridPP collaboration meetings) take place around three or four times per year. Other meetings include the integration teams' meetings, general developers' meetings, user board meetings, baseline working group meetings (responsible for gathering requirements from the experiments), technical coordination group (TCG) meetings (responsible for prioritising the requirements of the different communities being involved in EGEE) and engineering management team (EMT) meetings (dealing with daily development issues from an EGEE perspective) and a number of other sub-group's meetings. Representatives of the different projects involved in the GDC always take part in these meetings, which, as they argue, enables better knowledge socialisation within the projects.

Video conferences between groups of people responsible for each development task, or exchange of emails are the two standard ways of communicating within the wider GDC. Various mailing lists, such as the LCG rollout, the developers' mailing lists, TCG mailing lists, etc. provide a way of discussing and exchanging enquiries and solutions. Members of the GDC subscribe to mailing lists relevant to their own job function to keep up with issues raised, solutions proposed and sometimes directions for future work. In addition, wikis and instant messaging (e.g. Chat tool) are used for raising issues, making announcements, providing directions, discussing problems, finding solutions and gathering user requirements, etc. Such tools are believed useful in reducing the "overwhelming" amount of emails received and are seen as tools for *"fast communication in order to address short-term issues"*.

During meetings, either physical or virtual, priorities are set, new information is exchanged and issues are discussed (such as requirements, day-to-day development problems, bugs, and solutions, interoperability of components, new patches and new releases, user support and status reports). In such meetings, wikis and websites are used as records of previous communication and people openly share knowledge and discuss their work. Retrospective sense-making is an inherent and natural component in their systems development and it is clearly evident in these extensive communications; one can only realise this if they attend the WLCG meetings, or GridPP collaboration meetings where most presentations are retrospective and reflective and discussion and debates are centred more on what has been achieved since the last meeting rather than on the future.

It is worth noting that the communication between various management bodies and different groups does not only take place through documentations, even though minutes are taken in every meeting and are made available to the whole collaboration on the website. Rather, a lot of the communication happens non-hierarchically. In addition to this formal management structure of meetings and communications, most members of the wider GDC agree and acknowledge the importance of informal communication. As they stress, successful work often happens informally under informal face-to-face circumstances, e.g. over coffee breaks and meals, discussions in corridors, or by socialising in the pub. A middleware storage developer argued: *"There are ways to participate without being there physically, but the most important thing is that you meet people in the corridor, you meet people in the lobby and you interact more efficiently with people face-to-face…It is amazing the number of things you just pick up in the pub, because I know, oh I heard something interesting that for some reason hasn't come through the official channels of communication but just happens to come up, and it turns out to be a very useful piece of knowledge".*

Indeed the power of informal communications is commonly acknowledged with an LCG release coordinator even stating: "*I, as the release coordinator never forget the power of the informal requirement capture...PP and their computing has been around as a universe for quite some time and there are just a million interpersonal communications and channels, people who have worked on projects before together, people who associate their problems with one particular product, all this kind of stuff, and there is a remarkable amount of requirement capture which happens to these informal channels, and it makes its way to developers and developers can start implementing or passing software, even, to users, to see if this is the sort of thing they are interested in…and 3 weeks later we find somebody is reporting on bugs in something we haven't even released".*

Codification and documentation of the Grid's development in a structured way is not the primary goal of the LCG project (and GridPP), although wikis and websites are considered as stores of knowledge. While know-how is located and socialised through shared resources, more importantly, understanding and knowledge of various aspects of the project are embodied in key individuals, who are considered experts and carry out such knowledge and expertise to different clusters or groups of people by attending different meetings, sitting on various boards and constantly changing job posts. Furthermore, although most people in technical roles are employed for 2 to 3 years and there is a risk of losing expertise due to turnover, PPGDC believe that the structure of the project and the extensive communications foster the construction and sharing of knowledge through a high degree of socialising. Socialising with peers is considered by PPGDC as vital to ease the anxiety created due to the uncertainties and complexities faced (including the pressure of the LHC switch-on) and helps bond the group together: *"So having a team of people who are in the same boat as you where you can go out together and discuss about the fact that this bit of software doesn't work, or these demands which are placed on us by this VO are completely unreasonable and ridiculous is important. It fosters a bond between people and helps"*.

Improving social relations between developers when they meet face-to-face, with long coffee breaks and lunch breaks and foster socialising is therefore found by them to be important and useful. As an LCG physicist encompassing the role of the EGEE middleware development coordinator stressed: "*Having the possibility to have technical discussions all together is good. But having the possibility to have unstructured free time where people can talk to each other is also extremely important. This helps in building up the group. So we are trying to have these meetings that are very short, just 2 days every 4 months. So it does not impact much on the overall activity, in terms of a waste of time, but on the other hand I see they are extremely useful, and we normally have very big boosts after these meetings because we have the opportunity to discuss a lot of technical stuff, and also make people more available to others. Because sometimes when you talk just by email you feel the other person is just somebody who bothers you because they want something. When you meet the other person the approach is different".*

### 6.3.5 Consensus based decision-making and problem-solving

Decision-making on issues around the development, deployment and user support is based on a *"massive amount of talks"* and discussions where members of the wider GDC are invited to share their view towards reaching common consensus with fights and conflict being quite rare. Members of the GDC, including CSGDC, describe the decision-making process as *"a democratic process which can go around in circles and it is just by getting together with people and communicating with them, and putting forward an argument of why you actually want to do something that things are get done"*. As a computer scientist middleware developer stated "conflict resolution" is an inseparable part of this process and in the end the solution is *"a best fit rather than something which is ideal for one person in particular".*

PPGDC are characterised as an *"unruly crowd"* and it is argued to be hard to impose something on them. The lack of a formal management hierarchy, the distributed nature of the projects involved in the GDC and their alignment with each other and other projects means that there is a need to *"convince each and every person"*. Convincing and persuading the "crowd" is acknowledged as difficult and time-consuming but even CSGDC argue that: *"If you have a good reason for doing something people will always listen to what you propose, and if there is a good reason for doing it then you can generally get it done. It might take a while for that to happen, but they do listen".*

While the LCG (and GridPP) projects are based on "charismatic leadership" and, as PPGDC argue, the context of the collaboration is different than what is observed in industry, it is *"much more organic, much more consensus based"*, yet, common consensus is reached among GDC members only when people believe in something. As a former technical coordinator of the GridPP development effort and a latter GridPP project management board member stated: *"People will not work for you if they think what you are doing is wrong. So the only way they will do things is if we all agree that this is the right thing to do. There's no concept of a group leader saying, I direct you to do this. The only way that you can make things happen is by achieving consensus with a bit of direction where necessary. You'll never do it edict"*. The strong physics culture present in the GDC seems to mean that *"basically everybody gets listened to"* although not everybody's opinion has an equal weight.

Certainly sometimes there is disappointment as not all requirements or expectations get fulfilled, *"but this is just something they all have to live with".*

Decision-making does not appear to stem from social arbitrariness or political power. While political influence and vested interests exist, they still do not dictate the decisions made or what solutions should win through in the end. This is not to say that politics does not exist, but that they are dispersed, sidelined and the influence of powerful actors is dissipated. As an interviewee commented: *"There is a lot of vested interests and there is a considerably amount of politics between the different interests. Yet, nobody, no matter, even if they are the most politically powerful person in EGEE, cannot force a broken piece of software to be deployed because they will lose their political influence if they do that".* Thus, although, some decisions get made through the formal hierarchy (because the funding is dictating the structure or for other political reasons) and it may not be possible to internally reach an agreement, some decisions *"just get made by whoever shouts loudest, and whoever is paying attention that day".*

Indeed LCG's (and GridPP's) project management board appear to serve more in *"setting a direction for the people within the project"* rather than deciding how the project should move forward. Even EGEE, which is an EU and more formal project, cannot impose its decisions on people: *"the ultimate decision making is done by communicating with people and the ones you know from former experience. So whilst on the surface it is very formal, most of the decisions are still taken in this informal way".* Sometimes, however, conflict and differing opinions cannot be avoided and full consensus cannot be reached. People then have to find ways to compromise and as an LCG management board member indicated, this compromising *"can only be achieved by very extensive communication and frequent meetings on all sorts of activities".*

The following section discusses competition and funding pressures presenting a challenge in the development of the Grid.

## 6.4 Competition and funding pressures in the Grid's development

While PPGDC collaborate, they still compete and remain as individuals holding different interests. Competition does not only exist between the physics experiments however; rather, competition is more evident on a country level for Grid development. As the team leader of the LCG storage group described: *"There is a big war between the people in the US and people in Europe because they have a completely different budget and therefore they want to have their software used, otherwise they don't get money any more. In that case it's really competing teams, it's not collaboration"*. This competition generates various problems with the most critical one being the inability of different middleware components to interoperate. An LCG technical coordinator indeed remarked that: *"In the area of the grid middleware the diversity is much bigger. It is less under control somehow. It looks like different projects have complete freedom to do what they want, so at the end of the day you get the union of everything and you suffer a bit in that, because it takes time to find a way to evolve the system in the correct way"*.

Interestingly, funding is described by many as being the major problem. Firstly, there is competition because of the need to ensure funding, and secondly coordination of the GDC is not easy because it is difficult to define who is accountable for what, since different projects are funded by different institutions. For example, a GridPP deployment leader explained that: *"It is difficult to define who is accountable for what, because it is shared accountability... The complication is with EGEE because there is a funding agreement between our funding agency and EGEE and therefore we need to provide some of our effort making sure that we address EGEE's issues, as well as GridPP's and LCG's issues"*.

Most of the time the released software suffers precisely because in their attempt to secure funding people make unrealistic promises and release software with bugs that does not meet users' expectations. Funding also proves to be a barrier in collaboration and coordination of work because *"people see themselves working for a particular organisation and it is not that they do not like to cooperate with other people, rather they do not see it as their main goal, to work with someone in another country, funded by another project and maybe with different goals and different timescales"*. An experimental physicist/developer indeed stressed that *"people* *concentrate so much on getting the funding rather than to deliver what they have to deliver. This kills the project itself. It's wrong to fund such an enormous, huge project and especially when it comes to software development".*

Nevertheless, it is widely acknowledged that competition unveils better solutions with true quality coming from within: *"It's an open market and who eventually comes up with a better product wins. Now it might not be the best product but still it is going in the right direction. You cannot plan everything from the very beginning. So there is a certain degree of competition that is certainly desirable".* While EGEE, being a formalised project, tries to ensure that such competition is minimised, as they believe it creates a huge amount of overlap and wasting of recourses, it is still argued that without competition the work progresses slowly and brilliant ideas get killed off. Healthy competition is desired as argued by many PPGCD as it puts pressure on developers *"to make sure that their solution is the best"*.

Despite the challenges faced, PPGDC remain committed and try to be on top of things. While PPGDC acknowledge that there is competition between them, they however stress that the context of collaboration is much stronger and in the end their need for collaboration prevails. They somehow manage to make things work through lots of *"talking, and meetings and compromising"* and although they may seem to be constantly discovering problems and negotiating solutions, almost everybody has a strong belief that: *"The Grid will work; maybe not perfectly, but it will work"*. A significant source of this confidence resides in the belief in the skills, competence, pragmatic creativity and intelligence of PPGDC and in the formative context of collaboration: *"It will work for LHC, because we will make it happen...I have no doubt it will work, because we have immensely creative and talented people here to do that".* Indeed, as a physicist middleware developer commented: "*I am very confident in the abilities of the people in the project, on the focus of the project...on the goal, on our ability to work together as a team in the places where that is necessary and we will do it! Because you know we are smart people!"*.

Another source of this confidence (perhaps arrogance) resides in the PP community's long history of technological successes and an organisational culture which appreciates working with and around imperfection, with a physicist middleware developer indicating that: *"It will work and the reason is because there are a lot of people out there who are smart enough to know how to get it to work. So if there is a problem that comes up, people will stay up at night, or weekends, and figure out how to get it to work, and it may be a horrible, horrible hack, but it will work. The idea is you don't have to do this sort of thing in an ideal world, but I think if problems do come up then there are people out there with a lot of motivation and a lot of knowledge and a lot of skill who will come up with the solutions"*.

Having dealt with how the Grid development effort is organised and managed and with some of the challenges faced, the GDC's systems development practice is now described.

### 6.5 Dealing with systems development

Developing the Grid is described as a highly collaborative, distributed and dynamic systems development effort. Cutting-edge technology is used, new standards and protocols are negotiated and middleware together with other supporting software is developed in different countries and various programming languages. While the projects involved in the GDC face a number of similar issues to other distributed large-scale projects as described in Chapter 2, some of their challenges and characteristics are quite distinctive, particularly the scale of the Grid and the distributed nature of its own environment. LCG draws on several regional Grid structures in Europe (such as EGEE), US and Scandinavia, each using different middleware, something which raises issues of scalability, interoperability, standardisation and duplication of solutions. Even within the EGEE project, the middleware is modularised and its components are developed in a variety of programming languages. Although most middleware releases are tested in a small-scale pre-production system (PPS), they tend to be problematic when implemented across the whole system. The Grid therefore evolves as users actively engage in using, testing and reporting problems. In terms of systems development, simultaneous activities of development such as design, coding, testing, and maintenance exist, but also parallel solutions are developed and often compete with each other within the collaboration until one of them win.

The Grid is built on open source platforms (e.g. OpenGridForum software, Linux, etc.) and the way is developed shares similarities with open source activities. However, the difference from most open source projects is that it is not fundamentally about delivering a piece of software or a system, but it is about doing new physics. The system developed is obscured, it is complex as it sits on many machines, it is specific to the PP needs and the solutions adopted have to be agreed by the whole collaboration. PPGDC believe that their need for producing a working system by the tight deadline of the LHC justifies their need for experimentation and the ad-hoc development of the Grid, all distinctive characteristics of this environment. The Grid's development is performed in a highly collaborative manner exhibiting agility at a global scale. It is also developed with close involvement of the user community who exercise tremendous influence and pressures for the completion of a working system, which has to be achieved with limited time and resources. Further details about the similarities and differences of open source practices with the ones observed in this case are provided in Appendix A.

### 6.5.1 Improvisational evolutionary development

The development and deployment activity of the Grid is driven by the imperative to analyse data from the LHC. It is argued that the Grid technology being new and different precludes a plan-based approach to development. The aim of the PPGDC is thus to learn and move forward by "experimenting" through trial and error. Furthermore, the complexity, the pressures and scale of the projects involved in the GDC means that no-one can have a clear idea of the whole system; requirements cannot be pre-specified in detail; and even the development of the central Grid middleware cannot be clearly laid out beforehand and is therefore modularised and released gradually. LCG and GridPP pragmatically and creatively react to this, drawing on the down-to-earth and creative approaches embedded in the PP tradition and history, whilst EGEE aspire a more formal approach to development. As a PP technical expert stressed: *"The problem with Grid development is the nature of what we are trying to do because we don't know the requirements…it is certainly very hard to have any kind of formalised requirements, as people don't know, until they try it, what they are really getting and partly because most particle physicists, including me, are not trained in computer science enough to do these formal things. So the traditional way the particle physicists develop software, and what you see now* *in the PP experiments is just, by immediate feedback. So you don't try to develop formal specifications at all".*

The way PPGDC do development has been recognised by them as amethodical, pragmatic and improvisational. Traditional systems development methodologies usage is minimal with them claiming: *"How can you possibly do a formal software engineering approach to something that changes all the time? So software engineering is used up to a point but certainly not completely religiously"*. It is interestingly believed by them that the management methodology of software engineering practices is holding developers back from producing a technically competent and quick solution. While it is widely acknowledged that Grid development is *"a bit chaotic at times",* and it is described to be *"anarchic"*, yet, it is openly agreed that in some ways *"this is how it should be done". "There is only so much time until the LHC switch on"* and therefore PPGDC prefer the *"fast hack"*, creating a system which might be *"hacked together in some ways but it still works"*. Interestingly, what it is here observed is that developers are 'implicitly' allowed to hack because *"the Grid is still under development, the codes are under development"* and therefore this provides an opportunity *"to explore better, faster, niftier ways to reconstruct things, nicer algorithms"*.

The lack of formal processes in systems development is thus openly acknowledged within the wider GDC and this is because PPGDC believe it serves the prime purpose of building a working system in such a limited timescale. As a CSGDC claimed: *"[Physicists] are more pragmatic in computing...they are happier with an ad-hoc solution just to get the job done and push them through"*. PPGDC have often stressed the need for immediate acting toward problems and the ability to quickly cope with and adjust to situations that do not go as planned. Grid development therefore depends on improvisation and spontaneous agile actions in order to deal with problems and changing requirements. Being pragmatic and agile in performing development work is required in such an environment, which in some ways precludes employing common business methods such as quality assurance practices, the Unified Modelling Language, etc. It is interestingly argued by many PPGDC that such industrial practices are not designed to cope with this unique environment and therefore are found to be unsuitable, although CSGDC sometimes argue that formal methods are not used mainly because PPGDC "*just do not find it interesting enough to bother with"*.

While being aware or unaware of the more sophisticated methods and tools, PPGDC nevertheless make extensive use of other more flexible practices which allow them to be flexible and adaptable to the rapidly changing requirements, the external pressures and the tight deadlines of the LHC. As a computing coordinator stated: *"If computer scientists did this, they would take a much more theoretical approach. It might be rigorously more defensible, it might even be a better way of doing it in the long-term but in the short-term, you probably wouldn't get the results so quickly. And that's generally what physicists are concerned about. We want to know that when we get the data in December this year, there is going to be a system that works and can cope with it, and if there isn't, then we try to find ways to fix this in a very short timescale. I think it's that sort of approach which makes things work*". Being a new technological solution, the Grid presents challenges that require and somehow justify the continuous design adjustments and the need for agility and improvisation. Indeed a physicist middleware developer argued that the reason a formal process is not being put in place is *"partly because a lot of things move very quickly and thus you need to be flexible"*. As he stressed: *"The Grid is quite a different sort of concept. If the middleware goes down then no-one can use the system...therefore you have to react quickly, at times, to something that has gone wrong".*

While there is limited use of formal methods, still the systems development practices used within LCG and GridPP, broadly match with the general principles of agile methods: "individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiations; responding to change over following a plan" (Fowler and Highsmith 2001). PPGDC, therefore, see more value in producing a working system collaboratively by experimenting and by closely interacting with the users – the LHC experiments. LCG and GridPP focus more on people's creativity and competence rather than on methods. However, this evident agility becomes apparent at a global level (Zheng, Venters et al. 2007b) in an attempt to fulfil the requirements of a large-scale distributed GDC consisting of thousands of members. Their pragmatic systems development practice is indeed described by many as a bottom up approach to development, which is more about reacting to problems by writing a piece of code solving the problem and then fitting this code with the rest of the system rather than the other way around: *"The [development] approach has always been extremely pragmatic. So we are aware of a kind of a high level concept and a vision of what it should look like, but we always work bottom up, so we always start with primitive prototypes, leaving things out that are not necessary for achieving something, and try to get users involved as quickly as possible*". Extensive use of prototypes as well as the PPGDC's own experience is thus an important element of the development activity, which is argued to improve functionalities and facilitate requirements capturing, since physicists themselves know what the system needs to achieve.

### 6.5.2 Competing technological solutions

A distinct feature of identifying and exploiting technical solutions in the LCG and GridPP projects is the reliance on natural selection. Creating competing technological solutions is a traditional way of working within the PP community, since as a technical coordinator explained: *"Particle physicists are computing experts. If they need something, they will develop it themselves although it might exist"*.

While most of the middleware is formally developed by EGEE with people contributing from LCG and GridPP, it is still modularised and each of the components is prototyped, tested, released, deployed and improved in an evolutionary manner. Beyond this core software, there are often parallel technical solutions found in the LCG and GridPP projects for some of the core functionalities, such as some components of the middleware or other software packages developed locally to help deploy, monitor or manage aspects of the Grid. While, the drawbacks, such as interoperability issues, wasting of recourses, money and time of having four LHC experiments and the EGEE project each developing different frameworks and tools to finally do the same thing are acknowledged, it is argued that competing solutions provide motivation and crucial experience on how to develop the Grid. An LCG computing coordinator indeed stated that: *"We could not go through this effort of developing the Grid before any experience was gained. We needed to have several systems competing and then the best one would win and then that would eventually be adopted, become standard and included as part of the Grid"*. Similarly, an EGEE director commented: "*if you exclusively rely on one technical development and it fails, you can be in serious trouble - you've broken the entire grid, right? It just does not work at scale...so there is an implicit understanding that parallel development can be useful…the benefits outweigh the negatives".*

Different parallel solutions therefore compete with each other within the collaboration for a while until natural selection, e.g. because of technical failures, lack of funding, inability for future prospects, etc., (rather than just politics or social power) at some point define the one to be followed: *"The cream comes to the top. Things that work win through in the end and that's how we worked* it*".* This natural selection of competing solutions is widely acknowledged by PPGDC as a crucial way of identifying the most robust and efficient candidate solutions: *"People here are very pragmatic and the system that delivers it better will eventually prevail, and I think that is a good policy anyway".* While natural selection is used to structure innovation, it is, however, argued by a number of CSGDC that this leads to considerable waste and loss of morale from those not selected.

The Grid environment in this sense has been described to consist of a mixture of "ecosystems" in which multiple technical solutions co-exist and compete. The technical system therefore emerges from "contests of unfolding" (Knorr-Cetina 1999), so that the winning technology emerges as a fact of nature. Although political influence might be involved in such competition, it does not dictate the individual ecosystems. In this way the overall Grid emerges: "*you have different competing solutions trying to solve the same problem and everybody is kind of pushing their stuff, so in order for my bit to work with your bit, we will put glue in-between, etc....In the end you end up with something that is over-complex. But it works, it works at a cost, ok? But it works".*

### 6.5.3 Grid middleware development process

Although the LCG and GridPP projects have to align with EGEE, which means that they have to participate in detailed planning, something extremely atypical for the physics environment, PPGDC yet aim for short-term goals and therefore have short cycles of iteration with continuous releases. The processes involved in their systems development occur concurrently, rather than strictly ordered as in a traditional life cycle model or partially ordered as in a spiral process model and work is performed in a collaborative manner. The GDC co-evolves with the developer and user communities that reinvent and transfer software technologies as part of their team-building process. In this environment, *“the ‘customer’ has a much stronger position and a variety of choices”.* The development is driven by the PPGDC’s needs and even the development process is adapted to match the physics requirements which change constantly.

EGEE in their attempt to “*put a bit of method in the madness”* have tried to establish a formal software development process; a process for delivering, testing, and tracing software through its lifetime. For example, in order for any software to get into production it has to pass through certification testing and to be accepted as a production system and thus it has to go through a process *“that is tracing things, prioritising them and making sure that all the stakeholders are also formally agreeing with them”*. Every 8 to 12 months this process is seriously reviewed and adapted to the needs of the GDC. Nevertheless, it is still argued that the process is constraining and inflexible, and therefore it has, on numerous occasions, not been followed religiously by PPGDC, who still prefer to perform rapid development with immediate feedback, rather than to wait for their software to pass through a series of stages. A physicist middleware developer stated: *“The way we work is that we have a constant prototype in production. If something breaks we fix it immediately. We have frequent releases and every time we do a new release we receive a constant feedback from people. So it is not the traditional kind of cycle which is much longer, where you have time to collect user requirements, it is all about quick feedback”.*

Figure 6.1 below helps structure this section. It presents the Grid’s middleware development, deployment and user support process. It is numbered in order to reflect the different stages involved in these processes as described in the text. The purpose of this diagram is to guide the reader throughout this dense section.

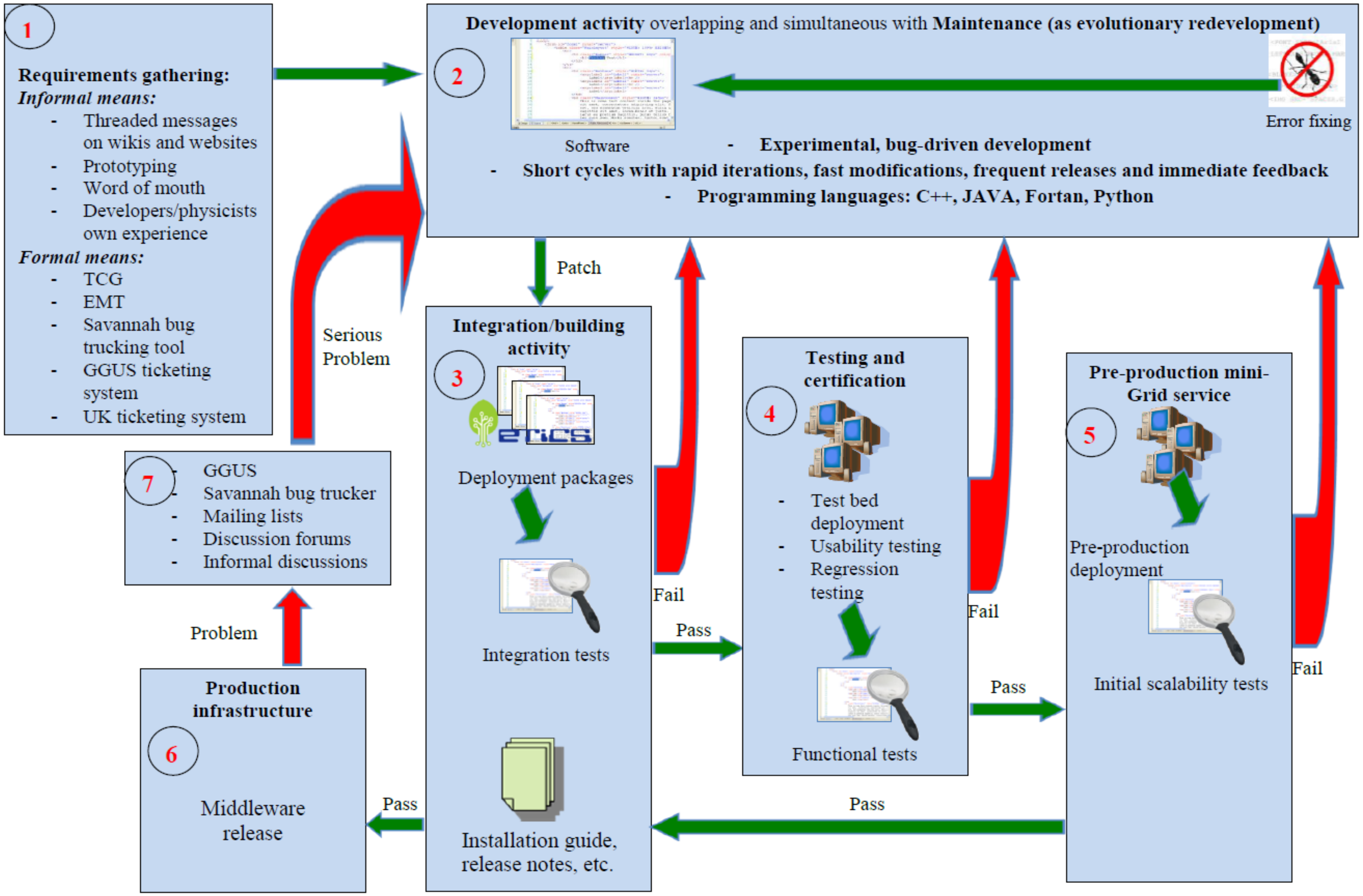


Figure 6.1 Grid middleware development process

***Requirements gathering and specification (number 1)***

The distributed development of the Grid presents a barrier to requirements capturing since experimental users cannot provide requirements for the final output in advance. The uncertainties and complexities of the GDC's projects are such that they preclude any formal requirements gathering and the creation of a formal specification. It is argued that on the number of times use cases have been attempted to be put together (mainly because of EGEE's persistence), this effort has failed because people cannot really agree on what the Grid is about.

Although CSGDC are eager in applying formal methods to requirements gathering, still the way initial requirements were captured was by developing prototype systems where users could play with and identify areas of improvement. As a middleware developer explained: *"The prototype was put out there where people could look at it to see what they could actually use it for. They would then start to try and build applications using it, come across a few problems, mention the problems and then we would sit down and have a think about how to actually solve them".*

Most of the GDC developers being particle physicists themselves have sufficient faith that they can develop something that serves the purpose of the PP community and reflects the needs of the LHC experiments and therefore they feel that this is a legitimate way to proceed without having to explicitly write down requirements from the experiments. A physicist middleware developer stated: "*the people working on the problem and those using the Grid come from the same background and thus they know what the problem is...For example, the users know the same as I do. So what we do is to construct the system in a way we think it should be done".* Immediate feedback is more valuable to them rather than wasting time applying formal techniques, which as they believe they *"cannot provide useful insights"*.

However, a number of formal channels for gathering requirements exist. The EGEE project gathers requirements through a number of routes from its key communities (including PP). EGEE, being a EU formal project has setup groups that are responsible for gathering requirements, setting priorities and discussing issues with developers and users. The first official channel for high-level requirements capture is the so-called TCG. Stakeholders of the different projects are present in the meetings of this group such as developers, people from the certification and testing teams, site managers and representatives of the experiments and other applications. The meetings of this group take place every 2 weeks and the minutes are discussed within the Tier-1 steering group, involving the deputies of each Tier-1 and the cluster leaders for each major activity. The Tier-1 steering group is mainly responsible for providing official directions to the developers of each Tier-1 working for EGEE, in order to prioritise the requests that come from the applications and create execution plans for various activities. As an EGEE manager explained: *"The TCG agrees on priorities, not just for development, but also for deployment of services. So LCG comes with a set of requirements, with their priorities, and that has to be put into all the other priorities, all the other activities that EGEE is doing. That is then given to the developers as a list of priorities. So there is quite a formal process of gathering that"*.

Another official EGEE channel for gathering and prioritising more specific day-to-day requirements is the EMT, which again involves developers as well as people responsible for packaging, integrating and managing the PPS. The meetings of this group take place twice a week. Discussions in this group are centred around the short-term prioritisation of the work and the important technical issues, such as what kind of patches are needed, what new developments should go into the next release, what bugs should be fixed depending on their priority, etc. The decisions of this group allow EGEE to have better control of the developers' activity. As an EGEE EMT member stated: *"We are supposed to work out how to implement the decisions of the TCG. At the same time we have to keep this maintained release rolling as well, and prioritising bug fixes and patches and stuff"*.

While the formal route for gathering requirements is fairly well-established within EGEE, the difficulty of gathering clear-cut requirements from the physics community is still widely acknowledged with EGEE members even arguing that *"in physics it is not quite as simple as that…they never have clear-cut requirements because they only know what the real requirements are as they move along. So the very standard process that we have – you gather the requirements, you put together the site, the site is being reviewed by the end-user, they say yes or no and then you move on, you develop the stuff and finally you present them with the final tool – that* *doesn't work in this academic environment because their requirements change with the tool. Only when they see the tool, do they know what actually would be possible and so develop it".*

Within the LCG and GridPP projects, the importance of informal requirements capturing is valued and openly recognised. In the PP community, informal requirements take the form of threaded messages or discussions on websites, wikis, mailing lists that are available to open review, elaboration, and refinement. Bug tracking systems (e.g. Savannah), and ticketing systems (e.g. Global Grid User Support System (GGUS) allow PP users to raise general enquiries, report bugs and put feature requests. These are useful since developers "*have a record of what people actually want and they can actually assess the difficulty in doing something, and the priorities for each of those requests".* More importantly however, it is the *"remarkable amount of requirements capture"* which happens over the *"million interpersonal communications and channels"*, informal meetings over coffee breaks, socialising in the pub and direct face-to-face interactions and which somehow *"find its way to developers"* which is considered crucial.

In this balance between formal and informal communications, it is the informal communication between developers and users that is seen as most important. The way the final output is shaped is not through formal routes of requirements capturing, rather through close interaction of developers and users. The importance of prototyping has been stressed many times, with PPGCD arguing that: *"You need to be in constant contact with the end user, you need to expose them to the early prototypes in order to make sure that whatever comes out eventually serves its purpose".*

Requirements analysis and specification in this setting therefore is *"an interactive process where the system is optimised by getting feedback as soon as possible and reacting quickly to problems".* They do not result from the explicitly stated needs of user representatives, focus groups or product marketing strategies. They rather are *"an uncertain negotiation with experiments"* and are seen as "*a general combination between day-to-day requirements and pre-conceived ideas based on past experience".*

***Experimental, bug-driven development (number 2)***

The actual development is typically experimental with no definite concrete goals besides adapting to new requirements and solving performance issues; it relies on an iterative process of fixing problems and is based on reusability and extension of already existing code with rapid iterations: *"Physicists do not often follow strict software design processes. They tend to just dive in there and write stuff".* When something is not working, it is *"chucked away and something else is used".* As a physicist middleware developer commented: *"We are not very formal in how we come up with the codes, for example, there is no real formal tracing of the design e.g. whether the design adheres to the architecture that has been defined. The architecture has been defined at the beginning of the project, but now, what is out there does not really reflect the architecture anymore".* The code is usually written in C++ and Java, because developers believe it is easier to pass best practices to other people through object oriented languages, although C and Fortran are still extensively used for developing some applications. Each development group can write code on various programming languages depending on their preferences and skill sets.

The development is highly bug-driven. Most development issues are tackled as soon as they are identified, bugs are filed for changing the design of certain components, and any discussions around development work usually result from bug comments. It is indeed argued that *"the actual process of development is by making incremental improvements to something running"* since developers *"never start with a clean sheet".* The development activity has fast development and feedback cycles, where an initial prototype is developed and based on bug identification by users, this is evolutionary improved with extra functionality. Every code change in the development is indeed made as a part of a "bug fix" for a uniquely numbered bug. The term bug is used to refer to any problems and any filed requests for modifications in the software, such as an actual defect, a missing feature, an enhancement, or a change in functionality. All change requests and their associated implementations have a unique number that identifies them. Bug numbers are used in this manner as a communication between developers who are free to exchange information between them through emails, chat, wikis and websites. Each bug has various properties attached to it, including the *owner* who is responsible for submitting the bug, a *summary* which is an one-line description of the bug, *comments* that describe the problem and discuss possible fixes, the *status* of the bug if it is unconfirmed, investigating, resolved, etc., the *priority* of the bug, such as how quickly it should be implemented, the *severity* of the bug which describes the importance of the impact of the bug to the system, e.g. critical, major and finally the person to whom this bug is assigned to.

Bug tracking tools, such as Savannah, are employed, through which different developers are notified of bugs filed on their responsibility. They are then responsible to schedule them according to their priority and severity. Developers usually also develop their own monitoring/testing scripts which allow them to find bugs in the system before making it available to users. The priority and importance of the bugs is usually discussed in EMT meetings, but also in informal discussions between developers and users. After a bug is fixed it forms a patch and it is decided, based on its priority, when this patch is ready to move forward for certification and then to either pre-production or to production. Most software in the development process is therefore provided as a package of patches and the full list of packages make up the so-called release.

Figure 6.2 provides an example of a bug report in the Savannah system.

**bug #15208**: *RB does not renew VOMS proxy extensions*

*Submitted by:* Maarten Litmaath <maart> Submit Changes http://

*Submitted on:* 2006-02-24 23:30

*Votes:* 55

| | | | |
|---|---|---|---|
| *Category:* | Job management | *Severity:* | 4 – Important |
| *Priority:* | 5 – Normal | *Item Group:* | Malfunctioning |
| *Status:* | None | *Privacy:* | Public |
| *Assigned to:* | David Smith <dhsmith> | *Open/Closed:* | Open |
| *Release:* | All | *Reproducibility:* | Every Time |
| *Effort:* | 0.00 | | |

Figure 6.2 Savannah bug report

The quality of the software is, therefore, mainly managed by this process of tracing issues and the resulting fixes, and by prioritising which components are made part of the release.

***Coordinated version control, system build, incremental release cycle (numbers 3-4)***
Software version control tools such as concurrent version systems or common centralised code bases, such as SourceForge, which allow changes to be developed concurrently and independently 'checked in' are widely used in the GDC. There is a single image of the code and at any time any developer can easily retrieve information. As a physicists middleware developer explained: *"I have put a case into SourceForge, to make sure that other people who are interested in using it can have access to the piece of code base and can further develop it. So by doing this, if for example I leave the job, I still make sure that there are other people who can still change the code and feedback on any improvements"*.

Such tools serve both as a centralised mechanism for coordinating the development and a venue for mediating control over which software enhancements or upgrades will be checked in to the archive. If checked in then these updates are made available to the rest of the developer groups as part of the official released versions, as well as of the daily build releases. Although such software version control requires coordination, because decentralised code contributors can independently contribute software updates that overlap, conflict or generate unwanted side-effects, it still allows synchronisation of dispersed and somewhat invisible development.

The release process of a software component was initially quite chaotic, something which created interoperability and integration issues between different middleware components. Interestingly, it was stated by a computer scientist member of the integration team that: *"In the beginning there was no build process. People would describe their code, by just sending their packages in a binary format and saying integrate that without providing any further detail. It was a nightmare. We could not work like that"*. EGEE therefore insisted on establishing a proper build process in place which was about building the different system components together in order to reduce dependency problems.

The e-Infrastructure for Testing, Integrational and Configuration of Software (ETICS) has been introduced to assist the software engineering activities involved in the Grid's development by providing tools and services to build, test and evaluate the quality of the software. Building and testing sophisticated software products, such as the Grid middleware, presents a challenge as it requires managing complex dependencies. Firstly it has to be built on many different platforms and clients or services located on the Internet in different sites have to be deployed; secondly metrics have to be collected and reports to be produced to help the certification team spot and address existing or potential issues. The ETICS system is used to manage this complexity by providing web-based tools and command-line application programming interfaces to describe the software, build it, produce packages for different platforms, test the deployment, and perform static, dynamic and functional tests: *"So the ETICS* is *a tool that takes all the different components, makes sure that they are compatible and builds them together, installs them together and things like that"*. Developers use ETICS to make local or nightly builds on their own machines or remote builds: *"So you can build your own code on your machine and test it, or you can submit a remote build, which is basically a build that goes somewhere on the ETICS infrastructure, is built and then the result is returned to you".* Different kinds of remote builds can be performed such as the "registered builds or volatile builds" which are seen as *"official builds, but ones which are not registered or used by others other than the real developer, and give developers the opportunity to verify that the build is correct and can be built on the infrastructure".*

After the successful completion of the build process the registered software products, which are usually bug fixes forming a patch, are forwarded to certification for further builds and testing including functionality testing, regression testing, deployment testing, etc. The patch acts as a means of communication between the development and certification teams as it includes important information such as: *"A number, a set of attributes, what are the ETICS stats that are needed, what are the configuration and also the release notes. So there is text proposed by the developer that explains what has been done with respect to the previous version".* The process of certification is thus seen as a process where further testing is performed but *"it is also about testing in a realistic environment, the environment of all the other things on the Grid…so there are lots of extra problems, and the certification really tries to duplicate that as far as possible"*. Indeed, as a member of the certification team commented: "*Normally we test the functionality but we also try to build the regression testing, so that every time you test something, you redo all the tests you have done in the previous phases; you basically verify that you are not reintroducing* *a problem that was cured in a previous version of the code and then, for some reason, it appears again. Then we try to do the deployment tests and the upgrade, downgrade to verify that nothing is broken".*

The successful integration and testing of the different patches means that these patches can now become a package that can then be forwarded to pre-production. Usually, when successfully certified, a new release to pre-production service is made every week. Every 3 or 4 weeks, each development cluster has what is called *"a developer's release"*, where the whole code is built, tagged and published for the wider collaboration *"to get all the changes of all the other collaborators in order to be able to move forward"*, while about twice a year, a 'production release' is created where everything comes together, is checked thoroughly for performance, and distributed on the pre-production grid and then to all the computing centres.

***Pre-production mini-Grid service (number 5)***

The successful certification of a patch means that certain criteria are fulfilled, different testing cases are successful and that proper documentation is put in place. The package is therefore considered as "ready to be shipped" and is moved to the so-called PPS which is seen as the *"last sanity check"* where experimental users as well as other independent people involved in the testing process can use the product and decide whether it is ready for production. It is argued that the PPS *"is supposed to be the first place that users get to see software before it is released. It is a big sort of deployment test for new stuff and it is a final opportunity to reject a patch because it doesn't work"*.

Pre-production runs like a mini-Grid service, where around 30 sites are committed to provide resources and facilitate and support users to run jobs and test the new functionality. As a PPS member explained: *"What we try in the pre-production is to have many different flavours of sites so that we know and we're pretty confident that it would integrate well in the local environment of the 250 sites that we have*". Any new package firstly gets installed to 2 or 3 sites and if initial testing is successful it is then installed to the rest of the pre-production sites for further testing.

Despite the high expectations for pre-production, in reality, the experiments and other applications are not committing enough to provide feedback. It is challenging to involve end-users at this stage as what users really require is *"to see the software running in their proper environment which is difficult to reproduce on small systems like the pre-production. So they prefer to have it rolled out to the production"*. Therefore, testing in PPS usually takes the form of "*kernels which are simple application tests which run more or less automatically in the pre-production to get sanity checks in"*.

Nevertheless, although some sites might volunteer to risk installing a new version of the system in order to provide feedback, this does not mean that the system will perform perfectly when in production: *"The scale is too small in the PPS. It is unlikely that the software will be fully debugged in the PPS...So the software, because of time scheduling or lack of volunteers or whatever else, just gets deployed out and everyone puts it on and you start to have a good experience or a bad experience, depending on the quality of the software"*.

It is indeed argued that most of the problems associated with pre-production firstly have to do with time as it is taking more than 6 months to be completed and secondly scale due to the inability to test the whole system and have a realistic view of its capabilities or inadequacies. Despite these, EGEE still requires from its developers, especially for components developed under EGEE, to make frequent releases to the pre-production service.

***Deployment and user support (numbers 6-7)***

When a new piece of software moves into production it is then up to the deployment team, which consists of representatives from sites such as system administrators and technical experts to provide directions on how the ~250 sites should install the software in their computing centres and support the day-to-day stuff for their end-users: *"so the deployment team should know which things are being released, know what problems exist, and they should coordinate the sites, upgrading and sorting out problems if there are any"*. The deployment process is an independent process as it is almost impossible to impose certain cycles on all sites, *"when for them the Grid is just yet another component of the whole suite that they install"*. Therefore, each site has the right to use their own update cycles to roll out new Grid services. They have their own schedule, their own maintenance procedures and only in the case of very critical releases can developers push out sites to deploy certain software by specific dates. Otherwise, they are free to choose how to deploy and when to deploy new software.

After the software is installed, there is a layer of communications and structures put in place for user support. If users experience any problems with the software, they can raise a ticket to the GGUS, or for those in the UK (as a part of GridPP) to the UK ticketing system, which are an entry point for users to report problems. Behind these systems, there are a number of experts who are responsible for investigating the problem and forwarding it to the appropriate person. If the problem concerns functionality, then a bug report is prepared and fed back to the developers: *"So actually the developers are not directly behind GGUS. The developers have their own bug tracking system that is something that they use for tracking their bugs. So a GGUS ticket, if identified to be a bug, is being converted into the bug tracking system of the developers and it then gets prioritised. They have a series of people who will look at the new bugs that are being raised and they will then allocate them to teams of developers who are responsible for the given code units"*. Various mailing lists also exist where end-users or system administrators can raise questions, as well as various discussion forums where other expert users can provide support.

The LCG and GridPP projects also run workshops and training events throughout the year which help train people working on the Grid and therefore cultivate user communities. But essentially users find solutions to their problems through interpersonal informal communications with other users and developers, either in face-to-face encounters or through discussion forums and emails. Interestingly, it is argued by an experimental physicist that: *"There is almost certainly, whatever it is you are working on, somebody in this building who knows something about it, if you can only find the right door to tap on"*. The first helping hand is always someone local who *"knows enough to help people get going"*.

Therefore, it is of critical importance to have expert people locally to provide user support. Developers are not necessarily considered ideal to do support as *"firstly they* *understand things too well and secondly quite often temperamentally are not necessarily suited to support. They would rather go off and do new things"*. They also can be very heavily loaded with development, and swamping them with user support queries distracts them from their work. For this reason, LCG and GridPP have cultivated the so-called 'power users' who are considered experts and can help facilitate users with their daily problems in order to allow the developers to develop. An LCG coordinator commented: *"locally, you need to have at least a power user who knows something about things locally who they can talk to. They can decide whether it is a Grid problem that needs to be reported, taken up somewhere, or whether it has to do with the actual experiment software. Because the experiment software is actually pretty complex as well".* However, one of the problems of power users is that while they know the *"quirks of the precise system they are dealing with, they don't necessarily have the overview to see how it should work...Sometimes they need that slightly broader view, to be able to identify the higher level problems"*. An LCG technical expert therefore stressed that: *"You have to be very careful about picking out your guinea pigs or power users. You have to have a few of them and they have to be very good and very friendly. For this we were lucky in some cases and very unlucky in others. But it proved to be quite helpful where people were very active in giving feedback"*.

***'Maintenance' as evolutionary redevelopment and re-invention (number 2)***

Software maintenance, which is about adding and subtracting functionality, debugging, restructuring and migrating across platforms, etc. is a widespread recurring process in the LCG and GridPP projects. This is perhaps unsurprising, since maintenance is generally viewed as the major activity of the development of any software. However, the traditional understanding of software maintenance does not fit with what was observed in the GDC. Rather, it is better to characterise the overall evolutionary dynamic practice of the community as ***re-invention*** or ***evolutionary redevelopment***. This re-invention primarily occurs when sharing, changing, examining and redistributing concepts, techniques and software that have appeared throughout the development of the Grid. It is here observed because the development of the Grid occurs through the evolution and iteration of prototypes to address the end-users' needs.

Re-invention is a continually emerging source of adaptation, learning and improvement in the Grid's functionality and quality. For example, as a computer scientist middleware developer stressed: *"How we have actually developed that, we built a prototype, to see whether this concept of Grid monitoring architecture would actually work. And once this was built and was out there and in production, all of a sudden it had to be more robust. Therefore we had to carry out continuous improvements, if it's just an enhancement or a maintenance, and we try to keep the cycles as short as possible and do them fast so that pretty early on we have them exposed to the end-user community and get some early adopters using it, even before we officially integrate it in the stack".* This re-invention takes place, in particular, while the Grid moves from a 'development mode' to a 'production mode'. The Grid thus evolves through minor improvements or mutations that are expressed, recombined and redistributed across many releases with short life cycles.

The developers themselves are continually producing these alterations by responding to reported bugs and suggestions for new features as well as by improving functionality guided by measurements produced by their various monitoring/testing tools. The alterations appear initially in their daily system builds and are then expressed as release versions that survive redistribution and review. As a result, these mutations adapt the Grid system to what the end–users and developers expect it to do while re-inventing the system. Indeed, a good example of this re-invention is provided by a middleware developer: *"At the moment we are in the maintenance section of the RGMA, as it is, on the production system now. We are now going through the redevelopment of RGMA. So we redesigned it, and we are now nearing the end of building it. So the maintenance is going on all of the time. And some of our time is spent fixing bugs in the system. So the head code that we are actually developing now is going to go on to the testing phase. We are testing as we go along, but it will be a major testing phase, and we have to actually test the interaction between the two systems because they will have to coexist together. So it all has to be backwards compatible".*

This evolutionary redevelopment therefore happens by adding new functionality to the already existing system, rather than by replacing the entire middleware component. It is not without problems, however, especially in such a large-scale distributed project. Firstly, it is inevitable that those involved in the maintenance of the Grid are mostly different from those engaged in its construction. Indeed, all the projects of the GDC, themselves, acknowledge that they are already losing excellent developers as the project ceases to be "exciting" and becomes "more routine". Therefore, it is widely agreed that knowledge needs to be well socialised in order for others to still be able to maintain the system. Secondly, the minor improvements need to be very carefully thought out when dealing with software that is already deployed and represents a 'working system' as this will most likely create performance issues and upset users.

## 6.6 Chapter summary

This chapter presented a detailed discussion on the collaborative distributed development of the PP Grid. Developing large-scale distributed systems is a complex activity that entails both technical and managerial challenges and this case has indeed demonstrated that. Particle physicists in this process of exploring with the unknown appear both unusual and somewhat traditional in the way they work. Freedom, trust, consensus, voluntarism, charismatic leadership, shared goals and internal motivation are all distinct characteristics of the PP community involved in the GDC and are seen to be its major driving forces. However, this is not to say that politics do not exist or that competition is minimised. Rather, healthy completion exists which helps bring competence in the community and blends expertise. Users play an important role in the development process, since being powerful they sometimes 'dictate' this process, but are also a major source of innovation as they are highly motivated and able to come up with creative solutions to problems and keep healthy competition going.

The key questions driving the rational of this chapter were explored and the following key issues have been identified:

*1) What is the nature of systems development for such a Grid project?*

Systems development for such a large-scale distributed project brought improvisation and emergent change in the forefront rather than methodological behaviour and planned change. It was here observed that the development process was more continuous, filled with surprise and more difficult to control. It was indeed realised that nothing goes as planned, agility and flexibility were required in order to quickly deal with unexpected problems and the development was mainly performed through competing technical solutions with natural selection declaring the most reliable and robust solution to be followed.

The management structures, communication channels and the decision-making and planning process of the wider GDC were highly influenced by the culture and history of the PP community's leaders and were such that facilitated and legitimised spontaneity and the improvised actions. Members of this collaboration were not organised in an industry-like way, as they argued, but in such a way that they brought about truly collective forms of working.

*2) What kinds of systems development practices are employed?*

The Grid is a new and emerging technology and the environment in which it is being developed is distributed, turbulent and uncertain. The LCG (and GridPP) projects' response was to respond with a spirit of pragmatism and agility. They improvised with the aim to improve by trial-and error. These fluid practices serve as a continuous response to change which together with the ad-hoc activities seemed to dominate the day-to-day practices. This study shows that the LHC Grid was developed in a constant negotiation between design and bricolage (Lanzara 1999), planning and improvisation, experimentation and discipline, and structured processes and amethodical practices.

*3) How is the development, deployment and user support being done?*

The development process has been thoroughly explored. In terms of systems development cycles there are not only simultaneous or overlapping activities of development but also parallel competing solutions of core functionality. Bug driven, experimental development and maintenance as re-invention and evolutionary redevelopment were found to be the most important phases of this process that occur concurrently, rather than strictly ordered. The importance of the various interpersonal informal communications is indeed highlighted as this is the most common way for gathering requirements and supporting users.

The following chapter presents a thorough analysis of the research findings against the background of AT which is employed to frame the work.

# 7 Distributed systems development activity

This chapter presents a thorough analysis of the research findings discussed in Chapter 6. Against the background of the previously discussed literature on collaborative practice and communities and on AT (Engestrom 1987), the aim here is to reveal the important interrelationships and contradictions within the activity system's components that characterise the collaborative construction and evolution of the Grid. This study's unit of analysis is identified as an overarching distributed systems development activity system – which is defined by the second generation traditional AT triangle, but, and in contrast to traditional AT, considers its achievement to consist of dynamically interrelated "networked sub-activity systems" – sharing the same object, tools, rules, division of labour and community. While this overarching activity system seemingly appears to be stable, tensions exist between the networked sub-activity systems and their elements that comprise it, which differ, contradict and compete in order to address their individual motivations.

AT's conceptual tools – the triangle model, the structure of an activity framework, the concept of contradictions and the expansive learning framework – are applied to the research findings in order to reveal those underlying dynamic relations that drive forward the overarching distributed systems development activity and surface important lessons learned that can inform other domains attempting to construct large-scale systems in a distributed fashion. Tensions and frictions throughout the development of the Grid are thus explored and surfaced. In particular, the focus is on how these are resolved through expansive learning cycles, and how they lead to the construction of new knowledge and the formation of new systems development practices.

The key questions driving the rationale of this chapter are: 1) What is/are the activity system(s) involved in the distributed systems development of the Grid? 2) What are the different viewpoints, interests and goals in relation to the artefact and the development activity? By analysing the viewpoints of the actors we find tensions and disturbances within the activity system and between activity systems calling for solutions. These contradictions manifest themselves as differences in priorities, understandings, expectations, motivations, etc. and in daily work practices which present a barrier to new forms of collaboration (Bertelsen 2003). A third question therefore is 3) How are these contradictions resolved in order for the activity system to achieve stability, facilitating learning and the evolution of systems development practices?

The chapter begins with Section 7.1 discussing in detail the overarching 'distributed systems development' activity system. In-depth descriptions of the different elements of the activity system are provided. Section 7.2 examines the divergent interpretations, motivations, interests, development styles and mentalities regarding the Grid's development. The conceptual tool of activity structure enhanced by Kaptelinin and Nardi's (2006) multi-motivation activity model and the tool of contradictions provided by the theory are used to surface contradictions and to explore how different attitudes are managed and lead towards the shared object. In Section 7.3 the expansive learning framework is applied to allow the researcher to identify lessons learned and describe the new collaborative systems development practices constructed as a response to contradictions in order to facilitate the distributed development activity. Section 7.4 summarises the chapter.

## 7.1 The historically developed activity system

Identifying an activity system (according to Engestrom's model) requires, somewhat simplistically, the identification of the following components: activity of interest, object, subjects, tools, rules, division of labour and community (Engestrom 1999a). In this case, however, the development activity is not undertaken by a coherent institution with clear rules and a division of labour. As the findings chapter demonstrates, various people, projects and institutions are involved in the Grid's distributed development – defined as the GDC – and in this setting, the end-users (experimental physicists) are powerful, developing software themselves sometimes in parallel with the actual developers, and heavily influencing, sometimes even 'dictating', the development process with their requirements. This therefore makes the Grid development activity of the GDC a complex activity, which requires us to take into account both the developers and particle physicists users who "act" as developers. The developers group consists of people from different communities and different projects (such as computer scientists and particle physicists some employed by LCG, GridPP, others by EGEE, and some even by all), already defined as the PPGDC and CSGDC.

PPGDC are defined as those particle physicists, both “developers” and users who “act” as developers, who are involved in the Grid’s development. The general term PPGDC was introduced for both PP groups when describing the findings for the sake of simplicity. Here, however, there is a need to distinguish between the particle physicists who are employed as developers by the different projects of the GDC and between the particle physicists users who are “acting” as developers developing parallel solutions simultaneously to the actual developers. This is required in an attempt to define the activity systems involved in the GDC’s development activity.

In interpreting the findings, three sub-activity systems are identified, which correspond closely to the different groups involved in the GDC – the particle physics developers sub-activity system (PPDAS), the computer scientists developers sub-activity system (CSDAS) and the particle physics users developers sub-activity system (PPUDAS). The three groups identified in the findings, are here theorised through AT as activity systems which interact, contradict and compete in order to achieve their goals. Here, therefore, the distributed development activity system of the GDC is presented as an overarching activity system, which is an amalgam of instances of a number of different networked sub-activity systems, which are all interrelated, yet at the same time compete in order to fulfil their priorities and motivations for developing a Grid that serves their purpose. The general term PPGDC will still be used throughout this section when discussing about both PP developers and PP users “acting as developers” involved in the Grid’s development.

This case indeed exhibits a network of interacting functionally linked activity systems that co-exist and comprise the overarching distributed systems development activity. As argued by Virkkunen and Kuutti (2000) the links between such networked sub-activity systems are determined by the objects and outcomes of the activities in the network. Engestrom (2001) has tried to capture and deal with this complexity by proposing a third generation of AT, which attempts to respond to existing criticisms to enhance the “dialogue, multiple perspectives and voices and network for interacting activity systems”. The expansive transformation of contradictions, therefore involve not just internal transformation within activity systems, but also transformations in the relations between systems. This third generation AT, however, remains under-theorised, with limited empirical evaluation of its effectiveness and limited literature particularly in the field of distributed systems development. This thesis, therefore, provides such much-needed evidence, demonstrating the value of complex networked AT and demonstrating its importance in a distributed context. Crucially however the thesis' adoption of the activity triangle (second generation AT) in structuring the analysis of such a messy distributed development situation leads to a new recursive structure for distributed activities which is defined as an "AT meta-framework 2.5" reflecting on ideas of the third generation. This meta-framework contributes to the third generation by enhancing the second generation of AT and forms a key theoretical contribution of the thesis.

Figure 7.1 below provides a diagrammatic description of this complexity with the first sub-activity system 1) being the PPDAS the second 2) being the CSDAS and the third 3) being the PPUDAS. The identified sub-activity systems are defined by their shared object. The first and third sub-activity systems belong to the same community and follow the same rules, tools and have the same division of labour. The focus of this thesis is on the Grid distributed systems development activity and more particularly on the LCG's project (and in a sense GridPP's) development activity with contributors from the EGEE project and computer scientists and therefore these sub-activity systems are only described from a development perspective. The user sub-activity system here (PPUDAS) is expressed as an instantiation of the development activity only because of the 'unique' power physicists end-users have being computing experts and that they have many times been referred to as "developers-users". The focus here is on the part of the user sub-activity system that is involved with development work. Thus, the focus of this chapter is to demonstrate the interrelation between developers and users and the impact users have on the development process, rather than to model in detail the overall user activity system which as a whole represents a neighbouring stakeholder activity (Miettinen and Hasu 2002) impacting on the overarching 'distributed systems development' activity.

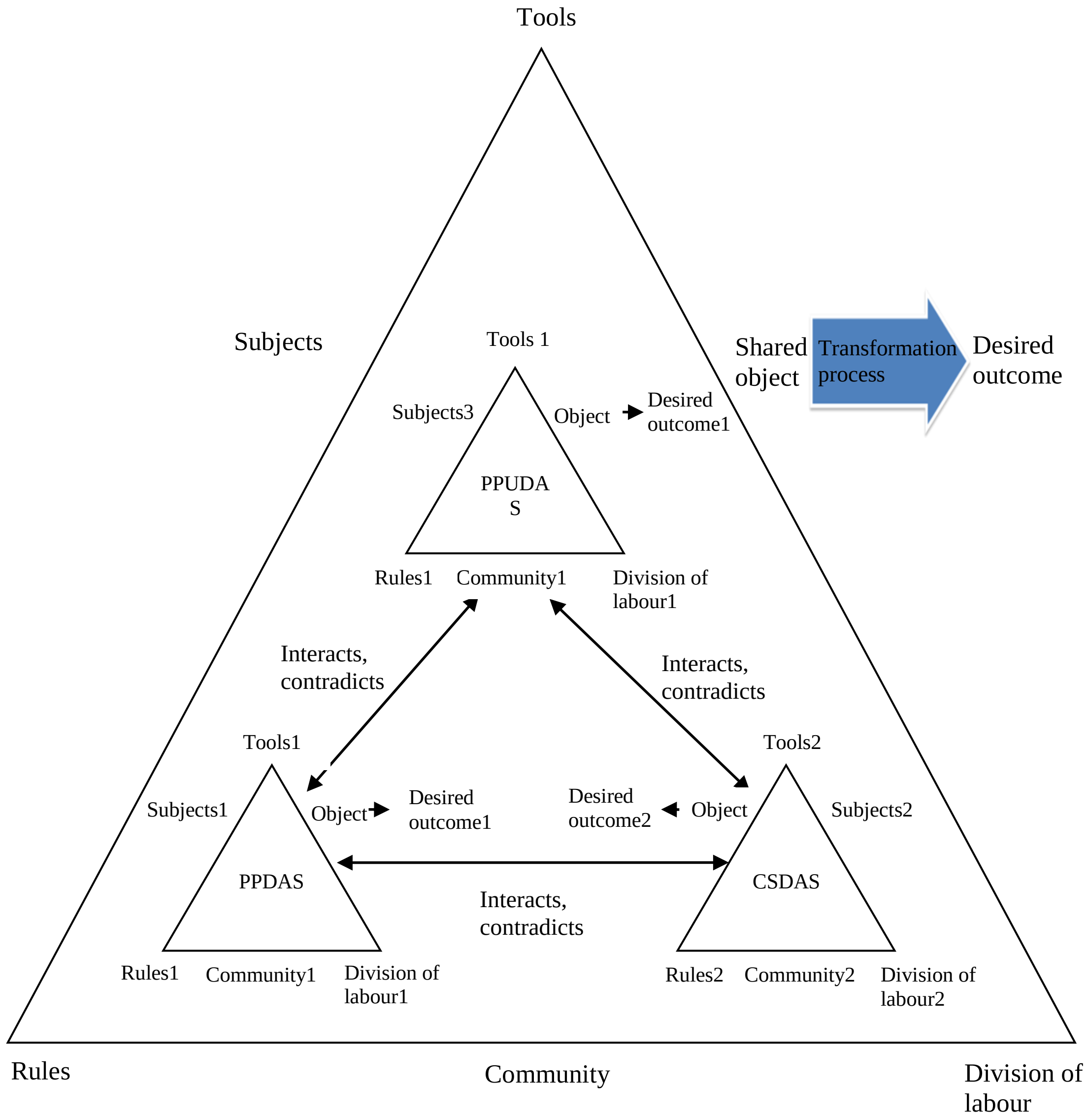


Figure 7.1 The overarching 'distributed systems development' activity system of the GDC with the networked sub-activity systems

The overarching 'distributed systems development' activity, therefore, presents a merge of these three networked sub-activity systems and demonstrates the dominant institutional practice followed which is the culture, mentality and style of development of the PP community. The tools, rules and division of labour of this overarching activity system present a combination of the tools, rules and division of labour of the three different networked sub-activity systems. The different sub-objects, while reflecting differing motivations and aspire to different outcomes (as they are linked to different projects e.g. the PPDAS and PPUDAS reflect the LCG's and GridPP's aspirations, while the CSDAS reflects the EGEE's goals), still merge to the one and only shared object of the overarching activity system which is the development of the Grid.

### 7.1.1 The overarching 'distributed systems development' activity system in detail

In this section a detailed analysis of the distributed development activity of the GDC is presented, identifying the elements of the overarching activity system. The object and desired outcome, the subjects, the tools, the community, the rules and the division of labour of this complex activity are identified in an attempt to create a rich understanding of the development activity and how this is influenced or shaped by the distributed collaborative context, the collaborative practices employed or emerged, and the community's inherent history and culture. The focus of this section is to analyse the elements of the overarching activity system and not to discuss about the networked sub-activity systems. These will be further explored later in terms of their contradictions. This overarching 'distributed systems development' activity system represents in Engestrom's (1987) parlance, the *central* activity in our analysis, which is the unit of analysis in this thesis; central because it constitutes the system in which the PPGDC together with CSGDC are the subjects, using collaborative practices and technologies to support the transformation of their object.

The elements of the overarching activity system are now discussed in detail, starting with the object and desired outcome.

***Object and desired outcome***

The object represents the intention that motivates this development activity and its transformation moves the subjects towards their goal (Jonassen and Rohrer-Murphy 1999). The PPGDC have been eagerly waiting for the LHC to begin to fully operate whereupon it will produce large volumes of data for analysis **(desired outcome1)**. The objective behind the LCG's (and GridPP's) systems development activity was realised when PPGDC understood that in order for the LHC to be successful they needed a distributed Grid to undertake their analysis **(shared object)**. This realisation made them form globally distributed virtual alliances with a number of actors such as funding bodies, universities and the industry, since the development of such a large-scale global infrastructure needed to be done collaboratively as a community effort. As one interviewee stated: *"What physicists want to do cannot be* *done by a small group, it needs a large collaboration"*. The development of this global infrastructure required people from different universities and institutes around the globe (including CERN) and involves projects from outside the physics community, including the significant contribution of the EGEE project. At the time, the EGEE project was launched to support a new form of science, called e-Science with the desire to provide a global infrastructure to support scientific work around the globe **(desired outcome2).** The Grid infrastructure was an appealing solution to facilitate this new form of science, however, developing such a global infrastructure was challenging and having two large projects (LCG and EGEE) performing parallel development work meant duplication of effort and waste of resources. This therefore, made EGEE realise that they had to join forces with LCG in order for this Grid vision to become a reality **(shared object)**.

***Subjects***

This distributed development effort consists of a global community of mostly PPGDC with technical computing skills and traditional CSGDC working together **(subjects)**. Different middleware components and applications are developed by both PPGDC and CSGDC collaborating together. The dominant culture and institutional practice followed, however, is that of PPGDC. This is particularly evident in the number of managerial positions and other key roles in the development activities and in the collaboration in general undertaken by physicists. PPGDC appear to be in a "position of power" and therefore appear to influence how work is performed. CSGDC are on short-term contracts, they do not usually have permanent positions as PPGDC often do and they are just employed to do the job. There is a feeling within the wider GDC that PPGDC's intellectual arrogance, and their pride of being physicists stemming from their computing history of successes, "allows" them to feel powerful and sometimes even more knowledgeable than others. An EGEE computer scientist stated: *"Particle physicists think their way is the right way because it works. And until somebody shows that the other way works better, then they'll carry on that way you know"*. A physicist's reply when asked about the Grid's success is interestingly an evidence of this mentality: *"If we cannot do it, then no-one can...Particle physicists will always get it done by and large, I mean history shows that"*.

A number of PPGDC have officially acquired the role of a developer for this activity by LCG, GridPP and/or EGEE; however, there are cases where actual particle physicists end-users (these are included in the PPGDC) mostly interested in performing scientific analysis, are involved in the development work, either by developing applications to run on top of the middleware, or by developing competing parallel solutions to the ones already provided. As an interviewee described: *"Some users are developers and some developers are users. There are people who concentrate equally in physics analysis and they develop the end algorithm for the analysis so those people are pure users. And there are users who are more of the application developer inside the experiments"*. PPGDC have indeed been characterised as *"powerful users, they do what they want. It's in their culture and mentality. If they need something, they will just go and get it or they will do it by themselves, although it might already exist"*. In the physics community, computing and physics are highly interlinked. The obscure nature of their work seems to suggest that *"no-one develops their stuff for them"*. In a sense they have to learn how to develop and use different technologies which as argued makes them want to be in control of everything and preferring to develop the Grid themselves, rather than use solutions developed by CSGDC or provided by other projects. It is argued by a number of CSGDC that experimental users *"need their software to be done very quickly"* and so they encompass the role of developers and do it themselves. For example, as an EGEE computer scientist commented: *"There are two competing solutions in CMS…but they still prefer to do it that way, just because they are a bit more control freakish and they want to keep more control and know exactly what is going on. And I think that is completely acceptable in this type of collaboration, that people do that. Nobody will say you have to use this if you want to get things done in another way"*.

The subjects involved in the activity, mostly the PPGDC, appear highly competent and it is argued that their level of determination and motivation and how well the group gets on, that are identified as the most positive and crucial aspects of this activity. Like in most professional domains, the subjects are expected to be self-motivated, good communicators and able to work in a collaborative environment. Indeed, as a senior LCG member indicated: *"There is a number of attributes you are looking for when you interview somebody. One, the technical skills and experience is* *a large part of it but probably only you know, one-half to two-thirds. I mean there's their ability to fit in a distributed team, the flexibility, you know, proactive problem-solving, the ability to work essentially unsupervised. Strong self-motivation and the ability to get on and do something, just because they are interested".* LCG (and GridPP) prefer to recruit people who are familiar with the history, social structures, culture and tradition of the PP community, and who are willing to step up and do the "dirty work" when necessary without leadership imposing on them or explicit reward, which sometimes conflicts with the CSGDC's mentality and the EGEE project's formalism. An LCG senior PP member commented: *"Particle physicists know the physics behind it, they know what the problems are and they interact frequently with the end-user communities, while computer scientists need more time to understand the project and the way the PP community works. And they want to do things very formally. They want the project to be very well-defined. But again, by definition, physicists normally don't know what they want".*

Commitment, devotion and voluntarism appear higher amongst PPGDC than might be expected in a commercial context. The PP leader of the storage group commented that: *"People are always really willing to learn in this community and take on advice; they enjoy working in the sort of academic environment where they get to work with very intelligent people all the time; they find it quite motivating. I think people are interested in the goals of the LHC – they want to see that experiments succeed and to find and discover new physics. Yes, I think that's the main reason that they want to stay involved in this sort of project".* While PPGDC certainly have personal career interests at stake, many express a sense of pride in working for a higher cause, perhaps explaining their willingness to undertake unpopular tasks when needed, although many PP PhD students could earn far more in working in the financial services industry.

The next element to be considered is the tools employed in the development activity.

***Tools***

The mediating tools of an activity as described by (Engestrom 1999b) can be anything used in the transformation process (physical e.g. a technology, or conceptual such as collaborative practices, programming languages, etc.). In this case, while certain technologies, such as Chat, wiki, etc. are extensively used to facilitate the development work, particular focus is given on the dynamic collaborative systems development practices of the subjects which in AT, such practices constitute an important form of mediated tools (Jarzabkowski 2003). Such collaborative practices distribute shared interpretations between the subjects, influence behaviour, predispose continuity and mediate between contradictions about the development of the Grid and therefore facilitate change.

The overarching systems development activity, while driven by differing motivations, is still highly influenced by the need of PPGDC to analyse data from the LHC. It is argued by a number of PPGDC that because Grid technology is a new and emerging technology, traditional systems development, with formalised plans, methods and tools no longer makes sense. The development activity here therefore does not follow strict plans and formal decision-making, rather it takes the form of exploration and spontaneity with the aim to learn and move forward by trial and error. The complexity, "multi-ownership", the pressures and scale of the project mean that no-one can have a full and clear overview of the system and therefore requirements are difficult to be pre-specified in detail, the infrastructure is developed based on assumptions and even the middleware is fluid and released gradually. The Grid itself is distributed and the development proceeds at different pace because of funding regimes. The response of LCG and GridPP to this is to pragmatically react, drawing on the down to earth and creative approaches and practices **(tools)** embedded in the PP tradition and history, although this contradicts with the industry "best-practices" ultimately wanted to be employed by the EGEE project management and CSGDC.

Practices such as rapid prototyping with immediate feedback, fast development with continuous releases, flexibility, improvisation, pragmatism and a user-driven development, are seen as effective and all serve the primary purpose of building a working system in an extremely limited timescale: "*You know, what the priorities are, heavy doses of pragmatism. We need this solved now. Yes, it is not the final solution, yes, it is an ad hoc way of maybe solving this problem, but if we don't have this solution now our production will fail for next year".* A bottom up and reactionary approach to development is followed with limited or no explicit use of traditional methodologies. Developers have short-term goals and therefore have short cycles of iteration with continuous releases. Releases are usually tested and certified before they move into the pre-production Grid, where further robustness tests and feedback are gained. The Grid therefore evolves through incremental changes, nightly builds and frequent releases.

The development is bug driven and the systems development practices used within LCG and GridPP broadly match the general principles of agile methods (Fowler and Highsmith 2001). Although PPGDC themselves parallel the way they do development with agile practices, they really do not know anything about them; they even argue many times that they might define their practices as agile practices such as XP because it *"looks fancy"*. As a physicist middleware developer stated: *"We do not stick to any one methodology. I suppose it is a similar methodology to XP, but it is not really following XP to the letter".* Defining their practices as agile, is considered as *"a matter of fashion".* The way they work somewhat aligns with agile principles, *"it is just that they have not figured out a different name yet"*. The practices employed stem from the PP culture and have been developed through a series of learning cycles leading to successful or sometimes less than successful outcomes. But this is how they evolve, get better and they can now deal with the problems faced. To cope with difficulties, emerging requirements and new opportunities, the distributed development activity has to be reactionary, flexible and quickly adapt to changes. Therefore, PPGDC believe that agility, ad-hoc actions and lots of experimentation are justified to be legitimate collaborative practices. This way of working is more appreciated by them than rigour and formalism, although this contradicts with CSGDC who aspire to a more formal approach to development.

A distinct *'tool'* of exploiting technical solutions in the project is the reliance on natural selection. Within PP tradition the development of competing technological solutions prevails – often as people simply try to solve their problems without consulting others. PPGDC posses an "aesthetics of imperfection" (Weick 2002), which sometimes upset CSGDC with their acceptance of the good enough. In this way however, and given just enough resources to draw upon, innovative solutions emerge and those with resilience survive. Once these "hacked" solutions exist, "natural selection" selects those to be carried forward; e.g. because of technical failures, lack of funding, etc., rather than politics or social power. This somewhat brutal approach is seen by PPGDC as necessary in ensuring that only the most relevant ideas survive.

Another form of mediated tools extensively used to support the distributed development work is a number of collaborative technologies such as wiki, websites, mailing lists, Chat, Evo (a video-conferencing tool) etc. that allow developers to keep up-to-date with all the requirements that come from the LHC experiments as well as to keep up with problems and codify solutions. Other tools aiding development such as, configuration management tools, bug tracking systems (e.g. Savannah), and ticketing systems (e.g. GGUS) are also used where users can raise general enquiries, report bugs and put forward feature requests. Such technological tools facilitate communication, coordination and control. While these are pretty an-advanced tools, they use them a lot. They have developed a way of working with these relatively simple web tools that not only helps pull the community together, but also helps hold the sense of community together in a much different way than the formal control type management seen elsewhere.

The next element of the overarching activity system to be considered is the community.

***Community***

The overarching 'distributed systems development' activity system of the GDC only makes sense, and can only be described, in the context of the community in which it operates (Nardi and Miller 1991). The community negotiates and mediates the rules and customs that describe how it functions, what it believes and the way that it supports different activities (Engestrom 1997). The community of this activity system therefore "refers to those individuals, groups or both, who share the same general objects and are defined by their division of labour, their rules, and shared norms and expectations" (Barab, Barnett et al. 2002). It is important to examine the community in order to define the nature of social interactions among the subjects and the beliefs, values and institutional practice that define or impact on their activity (Jonassen and Rohrer-Murphy 1999). The overarching 'distributed systems development' activity is largely situated within the PP community, which is widely dispersed. Although subjects of this community may simultaneously be members of other various communities as well, such as computer scientists being members of the EGEE community, the style of work and the mentality followed is still highly dominated by PP. PP practices, tradition and culture influence the way people collaborate and the way the Grid is developed. A computer scientist stated: *"The PP community and the goals and culture have had an enormous influence on how we [computer scientists] work... it is not that particle physicists are telling us what to do, it is just that we know what to do. It is the culture…You are being judged by physicists, right? You're saying does this meet the needs of physics? And at CERN the computing IT division is totally dominated and driven by the goals of physics, right? We [computer scientists] are very aware of who these guys are, they are masters. We work for them".*

Subjects of the community are continuously negotiating and altering their beliefs to adjust to the socially mediated expectations of the different communities they belong to (Nardi 1996a). Conflicts between the various roles and communities arise, leading to transformational activities to harmonise those contradictory expectations. However, although differing motivations and interests exist, the dominant institutional practice is the PP practice which is influenced by that community's history and culture: *"So by having said that, we are tied to this European project [EGEE] with a very formal structure; a lot of the work is still done, actually the successful work is done mostly in informal ways [influenced by the PP culture]. So whilst on the surface it is very formal, most of the decisions are still taken in this informal way. Which I think is important and intriguing".* Even the systems development process is *"itself very much dominated by the physics requirements".*

Proper software processes are not applied and this is because PPGDC *"need a working system pretty soon".* The PP style of development and way of working has somehow been ingrained into the CSGDC working in this Grid development effort and thus "forcing" them to work in similar ways. Indeed as a computer scientist technical coordinator commented: *"The bosses all come from the physics community, they are all physicists, maybe dedicated to computing but they are all physicists. So their way of working is also reflected in the way the computer scientists do the development work in the end".*

The PP **community** reflects some of the ideas of a CoP, a context within which learning and innovation occurs and is shared (Wenger 1998), and of colonial systems (Porra 1999); however it also presents distinctive characteristics (this will be further explored in the discussion section). The solid base of this community stands around collaboration (Engestrom 2000b) and this collaborative working is an inherent component of the community which traces back to their history, their culture and the nature of their experiments seen as collaborations.

A number of interviewees have characterised the PPGDC's collaborative systems development practice as not representative of practices particularly present in commercial environments. PPGDC are often surprisingly egalitarians and broadly international, they openly share knowledge and are happy to be dependent on each other and on the expertise of each other. The community is based on *"charismatic leadership*" and it has numerous times been stated that the context of the collaboration is what guides them: "*In our community, it is not that there is no hierarchy or no line management at all, there is. However, we are very much a community of equals, and we tend to get decisions made, not because I tell somebody in my group – you shall do it like this because I want it. It doesn't work like that"*. As the GridPP project leader commented: "*Particle physicists have trained themselves in the last five decades to work by collaborating. You cannot work in this field if you do not collaborate. It is important to have a reputation of being a good collaborator or you will not be able to do the physics".*

PPGDC argue that their aim is the pursuit of pure knowledge. They are passionate to do science and create scientific results and appear to be driven by shared goals and "sacred causes" and these shared visions, in their view are what minimises their individual needs. The common goal of doing physics appears to be an important source of motivation and provides a strong sense of direction, urgency and progress, facilitates a commitment to enable and support the collaboration, fosters a strong community bond, as well as binds efforts and bridges differences. Even the CSGDC are not "bored" contractors. While not driven by this "sacred cause" of PP, a number of CSGDC argue that they join LCG and GridPP because they are inspired to the PP goals and they want to work as part of the LHC. They are paid less or equal to commercial enterprises, however, they still like working at CERN and find the development of the Grid an interesting project to work on. Indeed there is something inherently cool about working at CERN. The researcher herself has experienced that. When for example the researcher talked to people about her research at CERN they became inherently more interested in the case. This is true for the CSGDC who argue that it *"sounds much better to work on a Grid project for CERN"*.

PPGDC almost always state with certainty that the Grid will work because they are extremely clever and will make it work: *"Within HEP, I think it's true to say that all people you deal with are highly intelligent and you don't get people working at this level who are not above average intelligence, given the general population"*. A more significant source of confidence (one might argue arrogance) resides in the belief in the individual skills, competence, creativity, and in the context of collaboration: *"It's just that we're amazing! You see it when you come to these meetings – everyone's a pretty smart person, the community is really good. You have all these smart people working towards the same goal and, at the same time, everyone is really friendly and quite willing to allow you to ask some questions to find out information. So I think there is quite a positive vibe within the community that kind of spurs people on to do a good job. Everyone's really approachable and is really willing to help everyone else"*. Indeed PPGDC appear to be creative people, who manage to find ways to deal with the Grid as imperfect as it is and it is in this ability to *"work around imperfection"* that others have faith in. This high degree of competence, the shared goals and internal motivation, the emotional bonds as well as the collaborative working are argued by them to create a trustworthy environment that drives the community. Trust is considered to be an important element that the PP community is based upon because it relies on people's commitment to do their job and on their readiness to make extra effort to ensure things get done: *"I'm very proud of particle physicists and I don't mind saying that. I'm proud of them because they always make things work"*.

The atmosphere of experimentation, trust, shared goal, and emotional bonds therefore appear to provide subjects with confidence to explore and make mistakes, with the knowledge that failures are legitimate learning experiences, and, when managed well, can ultimately contribute to the cause of the community. Most importantly though, the collaborative attitude which is nurtured over the years and is sustained by building a strong sense of belonging in the community, has proved to bring people together, although they might belong to different communities, and make them align themselves with the overarching ‘distributed systems development’ activity system’s object. While dedicated to the shared goal, this however does not mean that conflicts do not exist. For example, a GridPP physicist clearly showed his irritation when the monitoring tool he developed was dropped because the GridPP project’s management did not approve it.

Another important element of the activity system is its rules that are now discussed in detail.

***Rules***

Rules have been defined as the “specified acceptable interactions between members of the community” (Virkkunen and Kuutti 2000). Porra (1999) argues that the culture of collaboration or individualism within a colony is passed on from generation to generation as customs, norms, values, stories and behaviour. It is therefore “designed” into the community as ‘rules of conduct’. This is also the case for this PP community, since the collaborative way of working is inherent within individual PPGDC and it is a significant “real rule” that guides them **(rules)**. Being so distributed, it is important to have a strong sense of community and construct an identity for those involved in the Grid’s development in order for the project to function collectively: *“We have to work very well together as a team, in order for this project to be successful…and I think for us to socialise together is a very important thing*”. Going to the pub or going together for lunch, for example, are one aspect of it.

Traditional regulations such as those that might be expected in a commercial environment are not followed religiously in this community. Participant institutions are bound in this community by an MOU that directs how they conform. Development work does not follow a traditional plan-based management approach based on well-defined Gantt charts and fixed schedules. Whereas Gantt charts are produced in preparation for applying for funding, these serve as a minimal organising structure for the project, serving more as guidance as to how things should work. This, however, does not mean that they do not believe in planning. Planning is recognised as crucial for providing the fundamental legitimacy of the project and while there is not a clear fixed detailed plan, there is the plan to carry the project forward by improvising pragmatic and practical solutions.

Another form of informal rules and implicit cultural norms that guide this community's work is communication and socialisation. Such rules and norms are embedded in the way the meetings of different boards, committees and working groups, virtual meetings, informal face-to-face meetings, discussions on wikis, and mailing lists, etc. are organised. Although not formally stated, it is implicitly compulsory for subjects to get involved in such communications and socialisations of the community in order to have a clear overview of what is going on in the project and acquire a certain level of skill and knowledge since these are mostly acquired through *"word of mouth"*.

The last element of the activity system discussed is the division of labour.

***Division of labour***

Social structures and the division of labour are the ways in which people work together and organise their practices. Division of labour as defined by Cole and Engestrom (1993) is the "continuously negotiated distribution of tasks, powers and responsibilities among the participants of the activity system". It includes formal hierarchies directed by the power of positions held by certain participants, but it also includes less obvious structures – for example the way in which new members become acquainted, or the way in which identity is formed among the group.

Within the PP community there is no explicit **division of labour** and individuals can shift between jobs and have more than one job at the same time. Job titles do not mean that much. PPGDC volunteer to do things not because they are forced to by someone (as they argue), but because they want to feel that they have contributed to the 'cause' (though political forces can obviously play a part). There is huge reign of freedom and flexibility in roles because they tend to do what they feel they are best at. Such behaviour appears to be acceptable because history has shown that this freedom provides space for creativity and innovation. Furthermore, it is evident that there is no strict hierarchy within the community since, what they argue they have is a collaboration with spokespersons and volunteers rather than a company with a managing director and board. As an LCG coordinator argued: *"People are given autonomy to do things the way they want to, as long as they work, and it is not like the management breathe down people's necks saying you must definitely do this. It's a bit more sociable and amiable and people have freedom to express themselves"*. It is, however, clear that some roles and perceived hierarchies are imposed on the community by the outside funding agencies who insisted on roles like *"project manager"*. While for example in GridPP the "project manager" role was only filled because an IT industry representative sitting on the oversight committee persisted, and finally a particle physicist was appointed to the post, this role is now accepted as crucial to keeping the project on track. Yet the community ensures that such hierarchy does not dominate or become "managerial"; indeed there is considerable concern by some CSGDC that sometimes this hierarchy does not appear to be "managerial" enough and lacks the discipline of a company.

Despite the fluid roles, a number of different complementary, sometimes not pronounced, social structures are observed. The overall bureaucratic structure for the community's division of labour is represented as a "PP experiment". The community draws on the history and culture of the key leaders (all of whom are particle physicists) with a long tradition of large distributed collaborations and thus has structured LCG and GridPP like a PP experiment. There is also a somewhat clear hierarchy between the technical experts, with "senior" members demonstrating a greater ability to drive change and influence technical decisions. Such power is not necessarily mandated, but is often linked with the perceived expertise among the group, or the seniority in other areas (e.g. being a Chair within a university). Although such hierarchy exists, it is 'unspoken' and not defined in the projects' reports. Rather, it is an inherent component of the collaboration and exists because of the already established culture of respect.

It is indeed evident that while "management" is weak, there is a strong sense of community and identity and significant effort is put into maintaining such a sense of community. A relevant model for the community's social structure is Wenger's (1998) concept of CoP or Porra's (1999) colonial system, as this community shares some of their characteristics. Another social structure which is evident in the development activity is notably what is defined as the *"clusters of competence"* (which will extensively be discussed below); the way in which particular sub-groups become experts in a particular area of the Grid and co-locate (or emerge in one location) to reduce the significant difficulties of needing to constantly communicate with others undertaking similar work.

The community's division of labour appears to demonstrate a culture of equality with minimal structures; however, this does not ensure order or contentment. Many junior PPGDC and the CSGDC have numerous times expressed concern about the structure with a lack of process, a lack of predictability and a lack of knowledge of what is going on.

Having elaborated on all the elements of the overarching 'distributed systems development' activity system, these are summarised in Figure 7.2 below.

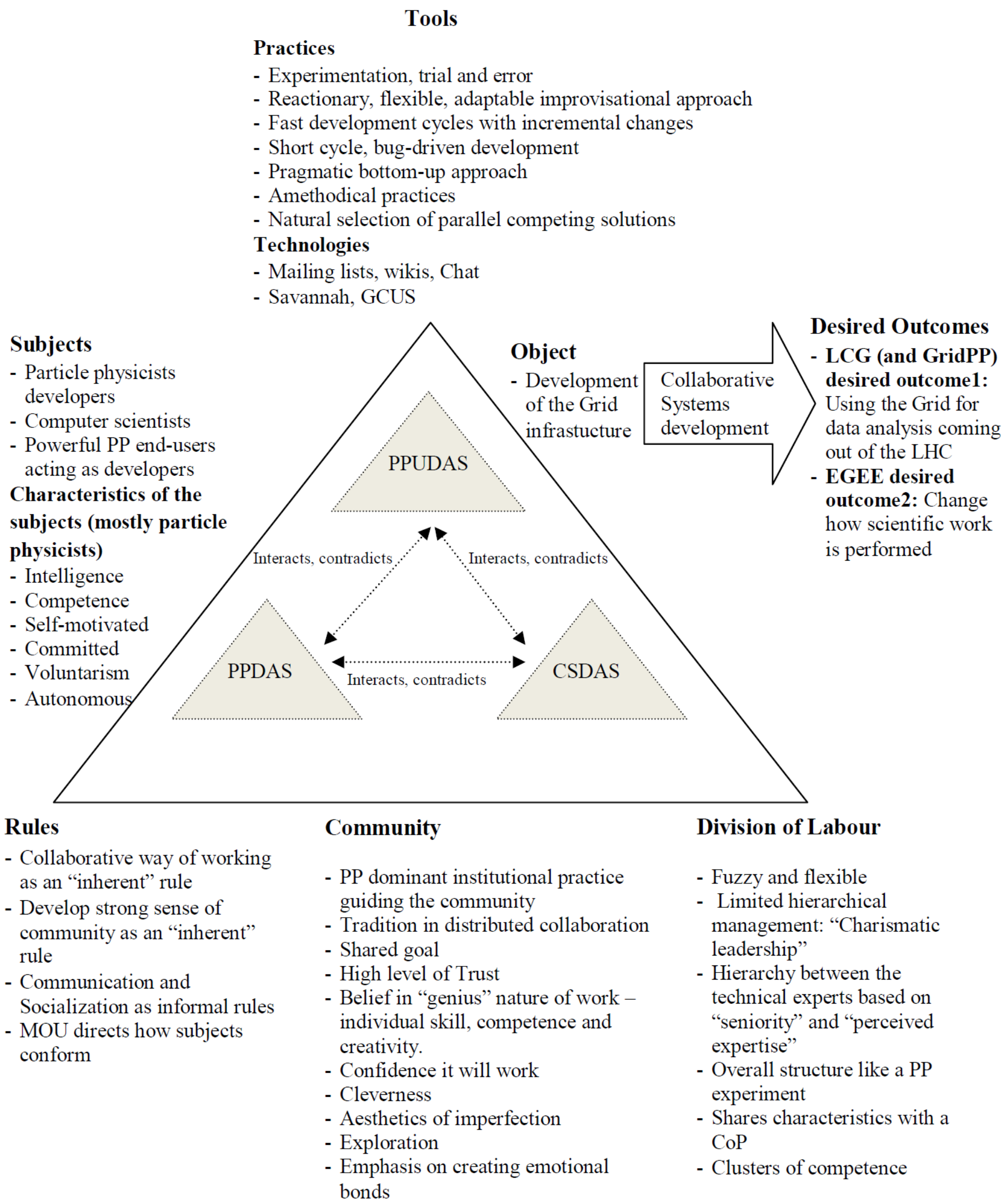


Figure 7.2 Details of the elements of the overarching 'distributed systems development' activity system of the GDC

The following section identifies the interrelations and contradictions within and between the elements of the overarching 'distributed systems development' activity system. These contradictions are manifested because of the tensions created between the networked sub-activity systems that comprise this overarching 'distributed systems development' activity. Such tensions are explored and thoroughly discussed.

## 7.2 The structure of the overarching 'distributed systems development' activity system and the internal contradictions

In the case of the collaborative distributed systems development of the Grid, a number of contradictions are observed throughout its development. Contradictions are fundamental tensions and misalignments that manifest themselves as problems, ruptures and breakdowns in the functioning of the overarching 'distributed systems development' activity system and present developmental opportunities (De Souza and Redmiles 2003).

A primary source of contradictions is when an activity is poly-motivated, thus driven by multiple and sometime contradicting motives, as explained in the theory chapter (Kaptelinin and Nardi 2006). This case represents an example of such poly-motivation, stemming from the networked sub-activity systems comprising the central overarching activity. The object of developing the Grid is thus defined by the whole set of motives the subjects try to achieve in their activity (ibid). This poly-motivation creates conflicts and disruptions that are not however directed towards the shared object, but rather to the instantiations of the object (ibid). Poly-motivation is not, however, the only reason for disruptions. Within the components of the central overarching 'distributed systems development' activity system there are continuous conflicts and transformations, which make the activity system continually reconstruct itself in an attempt to alleviate the pressing inner contradictions.

These disturbances, representing primary, secondary, tertiary and quaternary contradictions are now discussed in detail.

### 7.2.1 Tensions and fragmentations unveiling systemic contradictions

The focus of this section is to examine the overarching 'distributed systems development' activity system for any breakdowns, problems and tensions manifested between its elements and in relation to other stakeholders' activities. These contradictions exist because of tensions created between the networked sub-activity systems comprising the central overarching 'distributed systems development' activity system.

Developing the Grid is a challenging, complex and messy "political" effort. More than 2500 people from all around the world have come together, and committed to develop this grand next IT evolution. However, politics, funding pressures, competition, arrogance, different backgrounds and even the PP culture and tradition sometimes hinder the process. Members of this collaboration belong to different communities, different projects and they have different motivations of involvement, different interests as well as different expectations regarding the final output. On the one hand, developers have their own agendas while the experiments are quite powerful and want to "dictate" the development process rather than just provide their requirements. Even the two big projects involved, LCG and EGEE, are driven by different and in some cases contradicting goals. PPGDC have been asked to work with people outside their community (such as CSGDC) with different timelines and different priorities and it is argued that it is difficult and time-consuming to get people to collaborate and contribute.

The most profound contradictions exist on the one hand 1) on the macro-level between the two large projects LCG and EGEE and on the other on the micro-level between 2) PPGDC (including the PP users acting as developers) and CSGDC subjects among the development team.

**1) Tensions between LCG and EGEE**

***Divergent motivation for participation***

Both LCG and EGEE are projects with power and influence and have different motivations for participating in the development of the Grid. It is widely acknowledged that both projects are going in different directions, with LCG aspiring to the development and delivery of a working system for the LHC and EGEE, assuring the development of a generic Grid to be provided as a service to various communities and thus ensuring that all their partners are happy with what is being provided. As a physicist developing applications for the CMS experiment stated: *"EGEE is funded from the European Community so it has to show they are doing wonderful things and so on. In LCG it is not really like that. It's much more practical. So we are just focused on satisfying the needs of the LHC experiments".*

However, for the Grid to be developed it is necessary for these two projects to join forces, and share resources, knowledge and ideas. The LCG project existed long before the EGEE project was initiated, however, difficulty to acquire funding and bring expertise together, problems with coordinating such a complex distributed project, and the challenges faced when working in a distributed environment required this “partnership” to be created. An EGEE managerial member indeed explained that this partnership benefits all parties involved since “*EGEE offers one service, one common interface and people then specialise on top of that”*; This is what makes the idea so appealing “*EGEE being the common service and LCG building the specialisation for physics on top of that”*. A number of frictions have nevertheless been created between the two projects, because of the projects’ differing priorities and needs. As an EGEE representative indicated: *“There are so many people who are working towards a common goal but who are not working on the same project, and the problem is when priorities are different. And this is for example the problem between the LCG project and the EGEE project. They have very different priorities. And sometimes there are conflicts. For example, experiments complain because something they requested was not ready on time”.*

Having to ensure the development of a system that primarily serves the needs of the PP community reinforces the LCG project’s management actions of interfering with EGEE’s part of the development work and with the decisions regarding what software should be deployed. This leaves EGEE management feeling their freedom to be restricted in terms of what software to use in order to deploy its infrastructure. On the other hand, the LCG project in order to ensure “the quality” of the software used, means sometimes rejecting what EGEE provides and even sometimes using their own home-grown solutions. Certainly disturbances are created along these lines, since EGEE aspire for a Grid middleware to be used by all sciences.

Such poly-motivation behind the central ‘distributed systems development’ activity presents a *tertiary contradiction* between the instantiations of the object (Engestrom 1999a). The PPDAS (and PPUDAS) existed before EGEE and hence before the CSDAS was developed. The PPDAS (and PPUDAS) represented the central activity system at the time and they were motivated by a certain need that was to support the LHC in order to enable a breakthrough in physics. When EGEE joined forces with

LCG it meant that their motivation also had to be taken into account in order for the two activity systems to function together. The then central activity had to be remodelled to a *"culturally more advanced"* activity in order to take into account the differing motivations generated from the representatives of another culture (EGEE computer scientists) who introduced "culturally more advanced" motives to the central activity (Engestrom 2000b). The "*culturally more advanced"* activity, which is now represented by the central overarching 'distributed systems development' activity does not necessarily imply historical determinism, but it is seen as the "actual or potentially different way of conducting the central activity" (Bertelsen 2003).

The EGEE participants, as subjects involved in the *culturally more advanced activity*, were motivated by the generic transformation of the particle physicists' object. On the other hand, particle physicists, in responding to their personal and community needs of developing the Grid for the LHC, 'adopted' or seemed to 'agree' with EGEE's more generic motive at the beginning of the project to share the same outcome. While this adoption illustrated the harmony characterising the formative stages of the Grid project and ensured that the then central activity was entwined with the advanced activity encompassing the whole set of motives, throughout the development, the misalignment of the motives behind the object, guided a metamorphosis of the goals of the systems development actions of the then central activity to contradict those of the advanced activity (Kaptelinin and Nardi 2006). The seemingly harmonious inauguration of the project, therefore, stopped existing, with contradictions calling for the reshaping of parts of the now central overarching 'distributed systems development' activity.

Such contradiction is graphically represented in Figure 7.3 below.

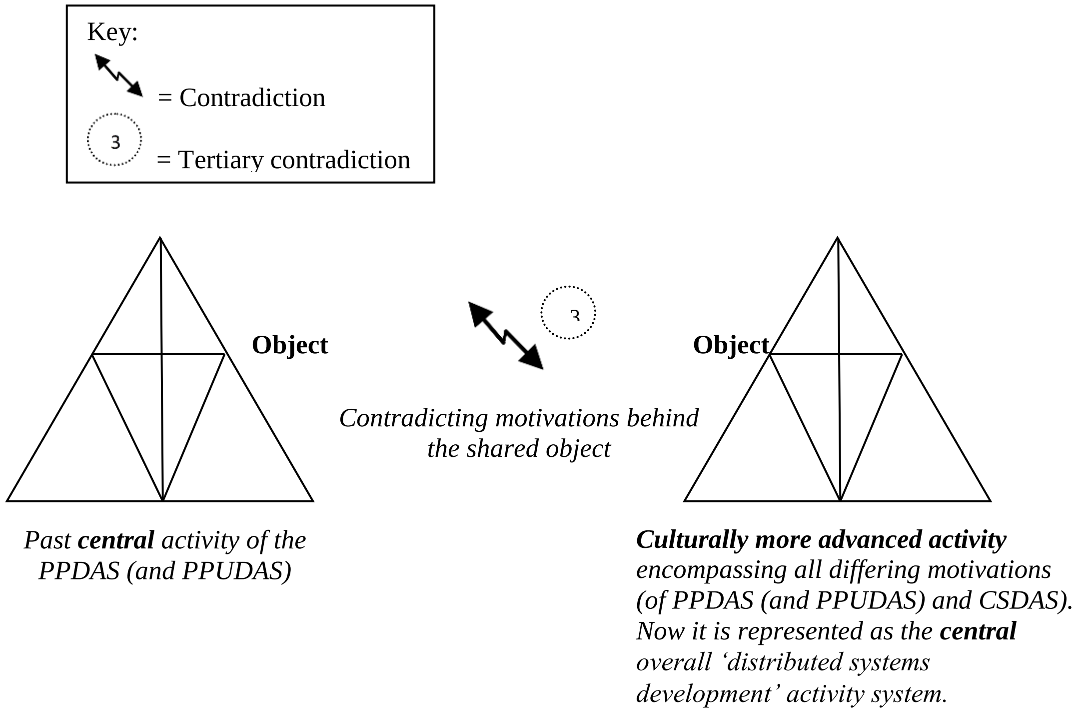


Figure 7.3 Tertiary contradiction between the central and culturally more advanced activity.

The central overarching 'distributed systems development' activity's object is therefore defined and shaped by the whole set of motives. Figure 7.4 diagrammatically presents the needs and motivations of the networked PPDAS (and PPUDAS) (N1 and M1) and CSDAS (N2 and M2) within a social context (CS) and under certain conditions and means (CM) that shape the shared object of the central activity based on Kaptelinin and Nardi's (2006) multi-motivation model defined in Chapter 3.

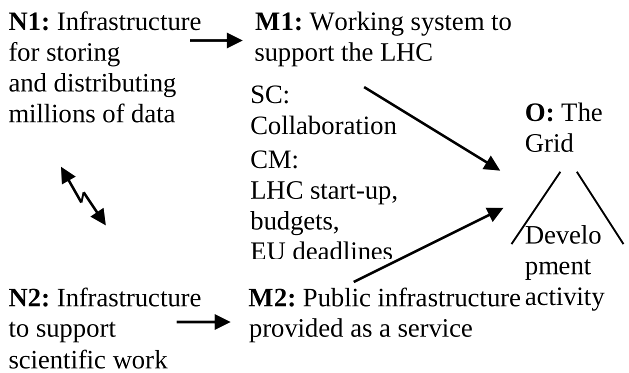


Figure 7.4 Poly-motivation within the central activity (adapted by (Kaptelinin and Nardi 2006))

### *A Grid for HEP or a Grid for all the sciences?*

The LCG project is providing most of the resources and manpower (developers) to EGEE and thus it has been numerous times argued that EGEE's Grid output is seen as the PP Grid, rather than a Grid to support all sciences. Finding a balance between serving the PP community and their other partners is indeed a challenge for EGEE. Being their largest user group, PPGDC, earn the right to be heard and therefore EGEE has to ensure that their requirements are fulfilled without, however, developing software which is too specific to address their other communities' needs. Tensions therefore exist around whether this Grid should be generic enough for other communities of users (which it is in part funded to be), or whether it should be tailored to PP needs (who are the major users and developers).

Distinguishing themselves from the LCG project is difficult, since the majority of the people involved in the development and as part of the EGEE project are particle physicists. This means that they know their requirements and what tools need to be developed to match with those requirements. An EGEE representative stressed that: *"A lot of the developments done inside EGEE are actually done by physics groups. So in a sense the shape of the Grid infrastructure we have, the facilities it has and its functionality is very heavily influenced by LCG and their view of the Grid infrastructure doesn't always match with the view we see from some other fields".* The challenge, however, is when other scientific fields need functionalities that are not essential for PPGDC. Intriguingly, while EGEE management worry about EGEE being locked into PP it is widely agreed within the LCG community that EGEE should be serving more its primary community rather than spending its effort on other communities. As an LCG physicist/developer stated: *"Our requirements are much more stringent. We need better performances, we need more functionality, that sort of thing. So if we can solve our problem and do it reliably, everybody else will get a real Rolls Royce product. Whereas I fear there is sometimes a bit of a rush to get new communities in before solving a lot of the problems".* Frictions are certainly created with the physics community mistrusting EGEE to delivering software for them since they strongly believe that their primary focus is not physics. There are even cases where PP sites refuse to install software which does not meet with their needs.

PPGDC have also numerous problems with the middleware provided by EGEE, as it is not working sufficiently for their needs. They argue that the software is not delivered on time and it is *"full of bugs"*. A physicist of the LHCb experiment stated that: *"The middleware received from EGEE is really far away from production quality. One of the complaints we have is that the middleware delivered is really a prototype, and we are trying to use it in large-scale productions as a real service".* Due to these problems, LCG have a number of times taken matters into their own hands; This means discarding EGEE components not specifically developed for physics and developing competing parallel solutions to what is already provided. An EGEE computer scientist indeed argued that: *"LCG really feel the pressure of having the LHC experiment that is starting shortly. And actually they have a number of developers working inside the experiment, which is even bigger than the number of developers that we have working on the EGEE project. So it is clear that for them it is possible to develop custom code, things that work only for them and work better than the code that is being provided by the EGEE project, because the EGEE is providing software that is supposed to be addressing the needs of all the clients, not just one*". The PP mentality is *"to react to what is in front of them"*, although this might mean duplication of solutions and a waste of resources, as believed by EGEE.

These conflicts and misalignments regarding the purpose of the Grid present a *primary contradiction* between the different communities which co-exist within the community element of the central overarching 'distributed systems development' activity system (Bertelsen 2003). Such tensions are indeed manifested because of the differing needs and expectations of the different networked sub-activity systems regarding the final Grid system. For example, the PPUDAS' needs of the Grid can be understood as emergent and having several dimensions. These needs relate to the major challenges emerging during their user activity and relate to the inadequacy of the Grid tools developed and provided by the community to support their work. For the CSDAS on the other hand, the Grid is just a generic tool to be developed and therefore if some specific functionalities are missing it does not matter, as long as the Grid works. The different sub-communities involved in the wider community element of the central overarching activity system, therefore, have different requirements and expectations of the Grid, something which presents a problem as the development cannot be straightforward, but requires successive refinements in order to incorporate all and sometimes contradicting requirements.
This primary contradiction is presented in Figure 7.5 below.

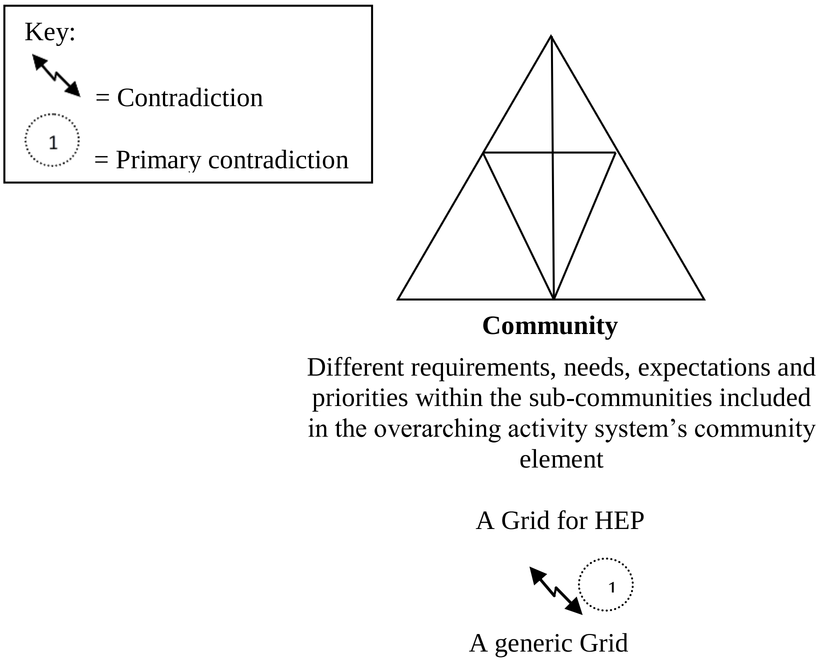


Figure 7.5 Primary contradiction within the community element of the overarching 'distributed systems development' activity system

***Divergent ideas on developing the collaborative Grid***

The differing motivations for participation also reflect the way the two projects aspire to do development and deployment of middleware components. EGEE is set-up as a formal project with a strong hierarchy, strong management and leadership, detailed timescales, well-defined deliverables and a formal development process with frequent releases, testing, etc. LCG, reflecting the PP culture, work in more unstructured, ad-hoc ways, with limited lines of authority and approaching the development as a bottom-up reaction to problems in their effort to create quicker solutions. Although EGEE aims for more structure and discipline in the development work, this is not enforced to LCG, resulting in *"dodgy, hacked, bypassed releases"*.

On the other hand, however, the EGEE's already established process a software component has to follow until reaching production is so lengthy causing delays and disappointment: *"EGEE deployment puts ridiculous processes and the length of time it takes from a piece of software being ready, as far as developers are concerned, to actually hitting a site is extremely long and is lengthened because so many people have to act on it – it's just crazy"*. This contradiction is indeed manifested as a *primary contradiction* within the division of labour component of the overarching activity system, with EGEE aspiring to a stronger hierarchy with well-defined deliverables coordinating the work, while LCG prefer to employ structures and practices reflecting the PP tradition; practices which have evolved over years of experience in dealing with distributed work. This primary contradiction occurs because of the contradicting division of labour of the networked sub-activity systems. The PPDAS and PPUDAS belong to the same community and share the same division of labour that is based more on "charismatic leadership", stemming from their PP culture, rather than on formal hierarchies. PPGDC themselves *"do not really like distinctions if that implies a boundary of what people should do".* There is flexibility in roles since people tend to do what they feel they are best at. On the other hand, the CSDAS, as a part of the EGEE community, need to necessarily define roles and responsibilities in more formal ways which conflicts with the majority's (being PPGDC) perceptions. Interestingly as a computer scientist stated: *"everybody is put in the same melting pot, there is no distinction between people and it is very difficult to reliably figure out who is working on what and which velocity, what is the quality that is going to come out of it*", which causes tensions between the networked sub-activity systems.

The primary contradiction within the division of labour element of the overarching activity system is presented in Figure 7.6 below.

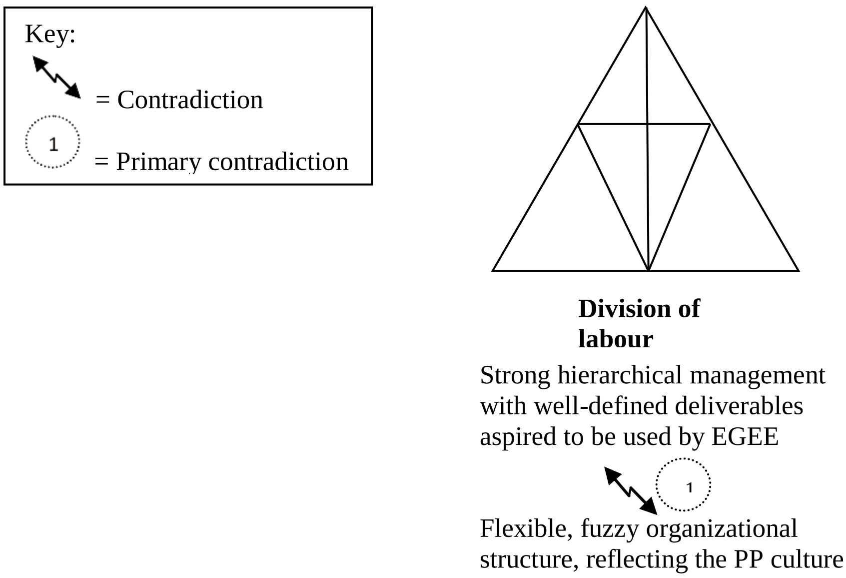


Figure 7.6 Primary contradiction within the division of labour element of the overarching 'distributed systems development' activity system

Another major contradiction is evidently illustrated in the different interpretations regarding how the Grid should be developed. It is argued that the EGEE project is very keen on using proper software engineering methodologies, while the LCG is not. Indeed, there is a strong belief in the PP community that methodologies hamper developers instead of helping them. Interestingly, as a particle physicist developer stressed: *"What you really need is a very clever and dedicated person who will go off and just solve the problem for you. The management methodology of software engineering practices can hold you back from producing a technically competent solution. And ultimately if you get someone who is very good, you don't over-manage them. You let them get on with it. And I think people feel that the management involved in software, formal software engineering ends up hampering the developers rather than really helping them".*

This view of how to perform development work causes a number of conflicts in the collaboration with EGEE complaining about the quality of the software products and stressing that: *"Computer scientists would probably solve things completely differently than a physicist would, for them [physicists] computing is just a tool, like a particle accelerator, to understand what happened in the Big Bang".* The LCG project and PPGDC are described by CSGDC as a *"bunch of clever people with a lot of experience in writing software"* but ones who are pretty independent and "slightly maverick" in the way that they develop software. However, as a GridPP interviewee indicated: *"at the end of the day, they [particle physicists] are the ones who produce the software that works".*

Although EGEE attempts to provide more structure and discipline to this Grid development effort (e.g. by enforcing a formal development process), it is nonetheless highly influenced by the PP tradition. The PP culture is the dominant in the EGEE project, which is sometimes found by others outside the PP community to be hindering progress: "*The biggest difference between the EGEE development style and LCG is because LCG develops software and experiments. And the experiments get into it in an extreme way, and thus the quality of the software you see around here is just disastrous. You do have to find a balance between the two. The quality of the software in EGEE in some aspects is extremely well conceived, but it does not solve the problem, so it doesn't really gain much, or you could argue which one of them is worse"*. The PPGDC research mindset, trial and error and ad-hoc, improvisational style of development contradicts the industry-like "best practice" techniques ultimately aspired to be used by EGEE management, something that PPGDC argue to be slowing down the process. Friction and tensions are created; however, CSGDC acknowledge that Grid computing would never have been at the state it is without particle physicists since *"there is nothing better than a hard deadline, like the LHC, to get something going"*.

Such different ideas regarding how the Grid should be developed present a *primary contradiction* within the tools component of the central overarching 'distributed systems development' activity. While on the one hand more flexible and improvisational approaches are aspired to be used by the LCG project and thus by the PPDAS and PPUDAS, the EGEE – and thus the CSDAS reflecting EGEE's aspirations – being a large, EU project aims for more formal structures and procedures resulting in an un-unified approach to development and to an output which sometimes leaves the user communities disappointed. The contradiction therefore results between principles, formalisation and specification on the one hand, and concrete practice on the other (Mwanza 2001) which occurs by the contradictions manifested between the tools of the networked sub-activity systems comprising the activity.

The primary contradictions within the tools of the overarching activity system are presented in Figure 7.7 below.

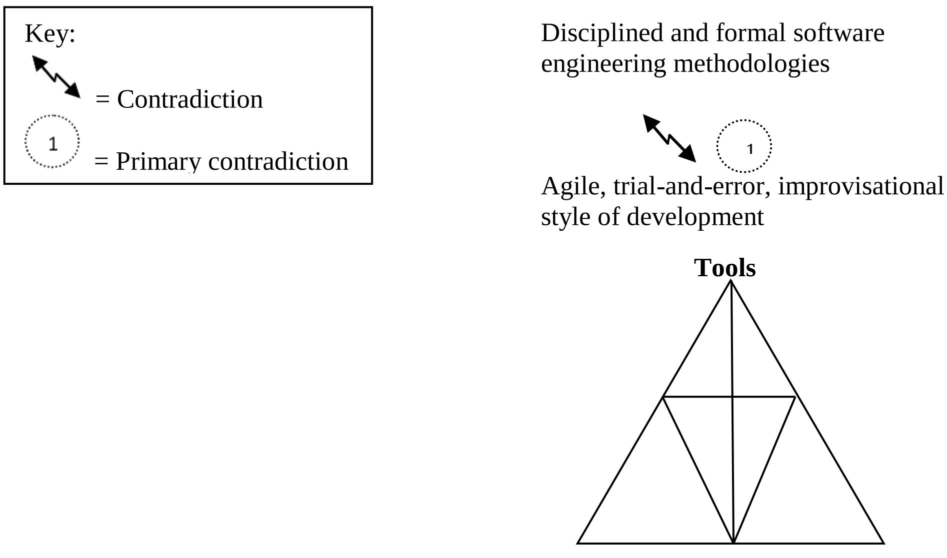


Figure 7.7 Primary contradiction within the tools element of the overarching 'distributed systems development' activity system

### *2)* Tensions between PPGDC and CSGDC

***Divergent motivation for participation***

Both PPGDC and CSGDC within the developers group of the central overarching activity system are motivated to become involved in the development of the Grid for different reasons. Contributing to this "sacred cause" is a strong motivation behind PPGDC's involvement, while for CSGDC working in such a large and promising Grid project could have a significant impact on their professional and individual careers and this is a more significant drive behind their participation to the LCG's effort. PPGDC argue that *"for computer scientists, this is just a job…they don't have the same motivation we [particle physicists] have".* As a particle physicist developer stated: *"I think the biggest difference is if they are not motivated [computer scientists] they simply care less about it. You used the word driven and I think that is true. And for many people on the physics side this is vocational and they couldn't imagine working anywhere else, this is the Mecca, this is why they get up in the morning. For other people, whether they worked here or worked at a bank, it doesn't make any difference. It certainly makes a big difference to me".*

Tensions between the two cultures are present within the developers group, representing a *primary contradiction* within the subject component of the central overarching 'distributed systems development' activity system. This contradiction is manifested between the motivations, beliefs, attitudes and expectations of the subjects involved in the networked PPDAS and CSDAS. On the one hand the contradictory interests of the two big projects, LCG and EGEE, present a challenge to PPGDC and CSGDC who are employed by both projects in their attempt to balance the differing requirements. While, on the other hand, the lack of the same motivation, not sharing the same passion and not interested in the same goals is hindering the development work with physicists even arguing: *"The problem with IT people is that they are much more interested in doing things the way they want to do them or there's a right way to do them or the most convenient way to do them, than actually understanding what it is that the community wants. So if an experiment asks for something, I would say you're much more likely to hear from an IT person, no, we can't do it because we do something different".* PPGDC know precisely the goals of their work. Being employed by EGEE or not does not really make a difference because as they argue their goal is to develop the Grid for the LHC and this transcends everything else. Yet, for CSGDC, this appears to be like any other job and it is argued by PPGDC that CSGDC *"suffer from a disconnect from the ultimate goal, which means that sometimes the software they develop does not do what it is supposed to do"*.

A senior member at CERN commented that PPGDC have a strong culture and history with special characteristics and it is this culture and history that forms the foundation for their community to survive. The PP world has been described as quite different from the computer scientists' world. Giving an example: *"If there is a new task that isn't described in the proposal for the project, that comes up, it is very hard to find people [computer scientists] to contribute to that. Whereas in a collaboration of the physics world, somebody would say – oh I can do that. It is much more difficult to find the effort to do things that are unplanned".* This difference in mentality, therefore, in their view, makes it extremely hard to hire computer scientists for the job, since they are not familiar with this kind of culture and way of working: *"You hire a PhD particle physicist, you know that when you hire them, they will come in immediately knowing what to do because it's sort of been ingrained in them over the last 3 years. But you hire computer scientists and it's a different culture. So this is a challenge not only for them coming in trying to understand how PP works but for us too".* This upsets CSGDC who feel that their skills are not appreciated.

The shared "sacred" cause that keeps PPGDC together, in their view, does not have the same effect on CSGDC. As a computer scientist argued: *"To build a community you need a common goal and a common goal usually comes from either, there is an important thing you have to do which is a difficult one, a difficult goal, so you are very committed to reaching that goal. Either there is a strong motivation from that point of view or from a competition with others to see who produces the best, in a broad sense, product. I don't think that we have any of these, most of the developers don't have neither of these".* Building a shared goal amongst the networked sub-activity systems therefore necessitates constant negotiations and consensual solutions in order for the different motivations to still lead to the shared object which is the Grid's development, either for the LHC or for the more general purpose which is for EGEE.

The primary contradiction within the subject element of the overarching activity system is presented in Figure 7.8 below.

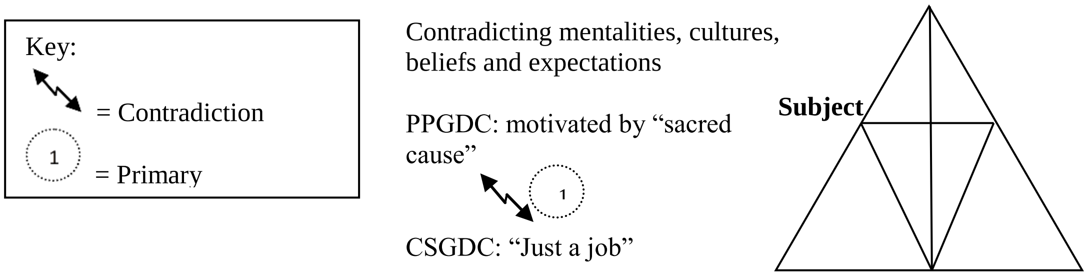


Figure 7.8 Primary contradiction within the subject element of the overarching 'distributed systems development' activity system

***Research experimentation versus development specifications***

The most apparent tension in the 'distributed systems development' activity is the necessity to balance research experimentation with formal development goals. As the Grid development effort progresses, the participants face an ongoing debate as to how precisely they need to define the project's components. PPGDC on the one hand argue for exploration and unrestricted flexibility while CSGDC are in favour of clarity and precise specifications. As a physicist team leader of a middleware component stated: *"Most people who develop come from a physics background and they tend to do more exploratory research kind of things; they try things out, see if they work well. It's less disciplined than industry. I have had people working for me who did software development in industry and they have found this environment very confusing and some of them have actually left because they couldn't live with it. We would say develop something and they would say what? They needed very precise specifications to work with and then they are very disciplined".* Indeed, PPGDC have been described as *"a marvellous species, who are not afraid of command lines and running things and are not afraid of exploring technical solutions"* with a lot of freedom in the way they work and that is why they feel that they should *"carry that freedom with them from their research into software development".* This, however, causes frictions with the CSGDC, who because of their training aspire to a more planned-based very well-defined methodological approach.

As PPGDC argue, too much specificity constrains the "discovery" intrinsic to research and the option to pursue new pathways as they emerge. Also, given the unknown elements in the development process, work does not necessarily happen as planned and in predictable time intervals. Tensions thus exist, as CSGDC believe that the research mindset of PPGDC hinders Grid development and slows production down, whereas, PPGDC insist that they are investigative scientists, exploring, and pushing boundaries and working with such a new and challenging technology like the Grid, presupposes exploration which can lead to new exciting pathways that are not already pre-planned and thought. The CSGDC's approach is to first *"learn how to work in an organised way and the actual knowledge about what has to go inside the software comes last"*, while PPGDC aspire to use a process where the content comes first. CSGDC find this style of development quite frustrating at times, with some even arguing that: *"There are no real boundaries...they [particle physicists] haven't really gone through a process. No, they haven't gone through that process at all. We have a few requirements that we haven't actually written down formally, or I don't believe that we have. Very often it is a reaction to what happens out there. Something goes wrong and someone wants to do something, then they start asking these questions. And then we have a look at what we can do and how we can best solve that problem. But there is nothing really which is formal in this design process"*.

Developing such a new and emerging technology like the Grid makes PPGDC legitimise the explorative nature of their work, something which presents a *secondary contradiction* in the central overarching 'distributed systems development' activity system between the skills of CSGDC being part of the 'subjects component' and the need for new flexible working practices (tools) for performing Grid development. This contradiction is manifested from tensions existing between the tool elements of the networked PPDAS (and PPUDAS) and CSDAS comprising the central activity. The mediated tools – practices – of the PPDAS (and PPUDAS) are more exploratory and improvisational in nature since they are based on skills and knowledge acquired from their PP culture of solving similar problems of "discovery-intrinsic" nature before. This, however, contradicts with the practices of the CSDAS that are more formal and well-defined procedures and are based on skills acquired from different development realities.

Such secondary contradiction is graphically represented in Figure 7.9.

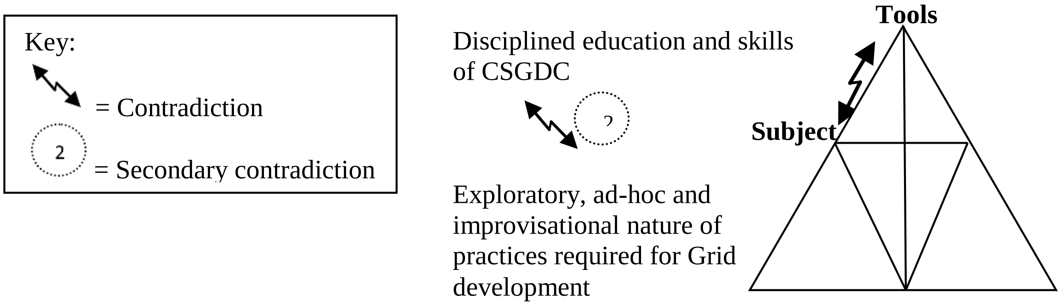


Figure 7.9 Secondary contradiction between the subject and tool elements of the overarching 'distributed systems development' activity system

***Well-defined systematic solutions versus working timely solutions***

PPGDC are satisfied with things that work although they might not work perfectly: *"Every physicist is a programmer. Now, the thing that the physicists are interested in is the physics and not the programming. So they don't care too much about the purity of the programme they have the aesthetics and the performance. What they want is something that works"*. Certainly they upset CSGDC among the developers group by their casual acceptance of the "good enough". It is argued that: "*The people who come from a computing background tend to have a slightly purer model of how the computing should work. And they get more frustrated with the perceived software deficiencies and they would like them to be solved in what they would term "properly", whereas the physicists are happier with an ad-hoc solution just to get the job done and push them through"*.

There is a feeling that CSGDC want to deliver something that is perfect although it might take longer to be completed. A working software delivered on time for the LHC is much more precious for PPGDC than a software that is nicely written. It is argued that: *"If you give physicists a problem to solve they will do something that works, and works moderately well, won't be very well written, but will solve the job. So they tend to get timely solutions but not particularly well-written solutions. If you give the computing people the same problem, they will do lots of design, they tend to get versions of it out there, for people to try too late, because they want to do a very thorough job in the first pass and it never quite emerges for people to use".* PPGDC fear the poor performance of the software written by CSGDC as they believe that *"computer scientists are interested too much in having nicely written programmes and finally don't care if they are usable or not"*. CSGDC on the other hand argue that the Grid middleware delivered is not the best quality they could provide; this is because the way it is being developed is by PPGDC aiming at *"the minimum necessary to make their day-to-day work as a physicist possible".* CSGDC stress that the most commonly used design patterns in the PP community is *"cut, copy and paste"*. They claim that PPGDC do not start development by defining clear specifications, rather people start working on old working pieces of code and try to adapt these to whatever they need. It is argued that *"the closer the LHC gets to switch-on, the more they [particle physicists] do quicker ad-hoc jobs to get it done".* Nevertheless, CSGDC fear for compatibility issues and difficulties when in production. A computer scientist indeed commented: *"We worked with a prototype and instead of actually chucking it away and saying – yes, we know, pretty much, what we want now, we will design this and put it together – it was actually put out into production. Because of this, we had to do the backwards compatibility…we are missing the kind of step that you would get in a company where you would say – thanks, that's a great prototype, now give it to the engineering team to turn it into a product; but here we don't have that engineering team doing this separately. Particle physicists don't have the experience to do it. So I think that is why it is hard".* Frictions and tensions exist with PPGDC mistrusting CSGDC and CSGDC "blaming" PPGDC for the development of a poor final product.

Such tensions represent a *secondary contradiction* between the object and the subject component of the central overarching 'distributed systems development' activity. While the shared object of the central activity is the development of the Grid infrastructure, the differing requirements and viewpoints of PPGDC and CSGDC within the subject element as well as their differing interpretations of the instantiations of that shared object present a barrier to development work and to the realisation of the final object (Virkkunen and Kuutti 2000). This contradiction emerges because of the contradicting goals guiding the development actions of the three networked sub-activity systems comprising the central overarching activity (Leont'ev 1978). The concrete goal of a working system that guides the development actions of the PPDAS and PPUDAS versus the goal of a well-defined carefully designed system of the CSDAS presents a challenge. This surfaces problems of mistrust and control that need to be resolved in order for the overarching activity to regain its balance.

This secondary contradiction between the subject and the object of the central activity system is represented in Figure 7.10.

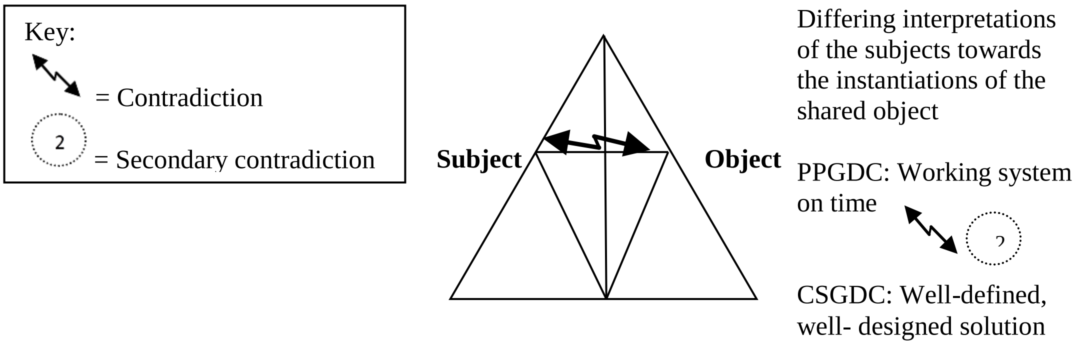


Figure 7.10 Secondary contradiction between the subject and object elements of the overarching 'distributed systems development' activity system

***Agility/flexibility versus rigor/discipline***

PPGDC feel confident enough with their way of working. With years of experience in distributed development, their pragmatic, practical, down to earth, bottom-up approaches are found by them to be "ideal" for this Grid development. It is argued that PPGDC are more agile and pragmatic than CSGDC who, as PPGDC argue are seen as "locked in" their top-down rigid and disciplined approaches to development.

Nevertheless, a change in attitude is observed while moving from the development stage towards the production stage. PPGDC acknowledge the need to provide a stable and reliant service to users and therefore try to change the way work is performed. It is now widely acknowledged that a change in attitude from a more innovative approach to a more risk averse and more structured and disciplined approach is required, with some even arguing that *"it is good to have a mix of both because then you try to get the best of both approaches"*.

Moving towards production makes PPGDC realise that in order for the shared object to be fulfilled and satisfy their LHC user community, a different attitude towards development is required. This presents a *secondary contradiction* between the object and the tools of the central overarching activity. This is evident because of the tensions manifested between the tools of the three networked sub-activity systems. A balance between agility/flexibility (evident in the PPDAS and PPUDAS) and rigor/discipline (evident in the CSDAS) is necessary and therefore for the shared object to be fulfilled – which is the development of a usable working Grid – the tools employed need to somehow embrace both styles of development, the innovative-explorative one, with the more structured and risk-averse one.

This secondary contradiction is presented in Figure 7.11.

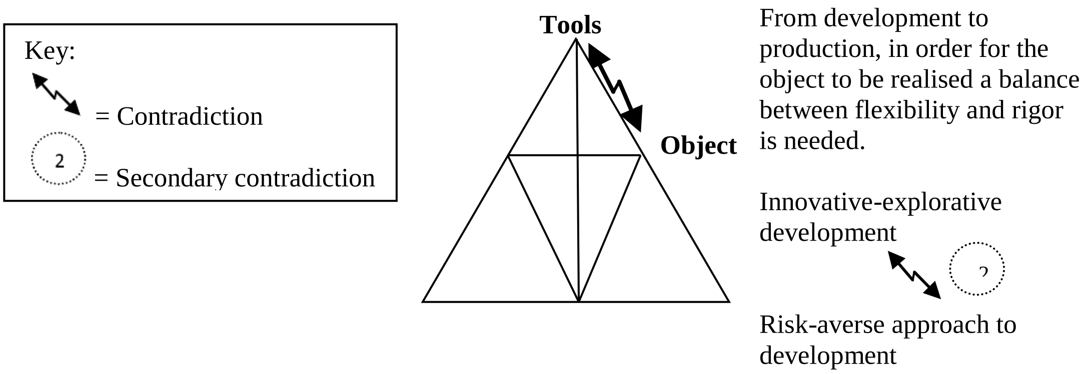


Figure 7.11 Secondary contradiction between the tools and object elements of the overarching 'distributed systems development' activity system

***3) Tensions between CSGDC and PP users acting as developers***

The Grid is already partially in use, and thus some physicists are already users who write programs to undertake their analysis. PP users (who are included in the PPGDC and from here on will be addressed as PP users) however, are powerful users since they are computing experts and therefore they prefer to write their own code rather than use something that although more generic already exists. It is argued that being technologically competent, PP users can work around imperfection in many different ways and can develop solutions that work to match their requirements. Therefore, when asked what they would like the system to do, they *"tend to give technical implementations as a response, rather than a requirement"*. As an EGEE computer scientist indicated*: "There are a lot more of the particle physicist users than there are of us. So if they have a problem they can go and solve it themselves, rather than coming to us and saying –why don't you add this into your code for us? There are just so many of them. If they don't like what we have done they can go away and write something else".*

PP users are not passive in this development process, but are active agents responding to the Grid as a part of their practice in their attempt to create specialist interfaces tailored to the specific needs of their practice. They tend to mistrust anyone else that tries to do the job for them. This however, leads to duplication of solutions, which as they acknowledge are not without cost both in wasted effort and resources and poor maintainability. Indeed CSGDC are upset with PP users trying to "become" developers, but PP users believe that this "natural selection" of solutions ensures that the most relevant and resilient ideas survive.

PP users, being reluctant to install anything that might create reliability issues in their existing system, prefer to write their own solutions. CSGDC argue that PP users believe *"they can do any kind of thing...software is one of these things and they think they can do it better".* CSGDC are however upset with this behaviour, but they also realise that they cannot do anything to change this or prevent it. They however stress that: *"The problem is when the good stuff comes along, and other stuff is already working and it is very hard to make the effort to make users change to something that is going to do pretty much what their existing stuff is already doing".* They thus criticise the PP users' "natural selection" argument.

Being powerful users, particle physicists, like to feel in charge of their work and be able to do their analysis by using the fastest route available. Certainly, to use the Grid, PP users must see it as the fastest route. The Grid is just a means to an end. As an experimental physicists argued: *"Our primary purpose is to analyse the data, if we can do it on the grid, that's fine, but if it gets in the way I am sure we'll just chop it out"*. They have on many occasions targeted parts of the Grid to run their jobs which are more robust, something which creates tensions with CSGDC as they would like them to use all parts of the Grid in order to test them for reliability. An EGEE computer scientist provided a nice example of this situation: *"The middleware we have developed within EGEE, most of the data management stuff is not used. Because the experiments want to do it their own way. They don't like the management system we have produced. Because it has not been very reliable…they have found that the grand vision that we had was to be able to find the best resources and submit the jobs to those resources, according to where the data were. So what they actually do is, they submit the so-called pilot jobs, and once that job has found a machine where it runs properly and has the right software installed it just sits there, running and replaces itself with another job, which it gets off an experiment managed queue".* While CSGDC aspire to creating the machine of the future, a "coordinated resource sharing" Grid, PP users see the Grid as just a tool to get the job done and if it is not working sufficiently to their needs, then they will develop something else in order to replace it. PP users do not like waiting for CSGDC to provide support. This frustrates CSGDC who feel that their work is not appreciated.

This user interference in development work causes frictions and imbalance within the subject element of the overarching 'distributed systems development' activity system and more particularly between CSGDC and PP users. Such *primary contradiction* is observed because of tensions between the subject elements of the networked PPUDAS and CSDAS.

Such primary contradiction is identified because of a quaternary contradiction previously manifested between the object of the central 'overarching activity system' and the tool element of the neighbouring, co-existing and concurrent activity defined as the 'overall user activity system'. The PPUDAS, which belongs to the central 'overarching activity system', is an instantiation of the 'overall user activity system' and is defined as only the part of the 'overall user activity system' that deals with development work. The 'overall user activity system' represents a neighbouring activity to the overarching activity system and therefore the problems, tensions and misfits that arise between the two co-existing activity systems represent a quaternary contradiction. This quaternary contradiction is observed because the Grid, which is the object of the overarching activity and represents the tool of the 'overall user activity' is problematic. In this case the subjects of the 'overall user activity system' (the PP users – who are the subjects of the PPUDAS) embrace the role of developers in order to address these problems. This therefore creates a contradiction between the subjects of the PPUDAS and CSDAS, which manifests itself as a primary contradiction within the subject element of the overarching activity system. The computing expertise of PP users, in their view, "allows" them to "encompass" the role of developers, something, however, which contradicts with the traditional nature of systems development and challenges the fixed meaning and traditional interpretation of the roles of 'system developer' or 'user' as understood by the subjects of the CSDAS.

The primary contradiction within the subjects element of the central overarching 'distributed systems development' activity system – resulting from the quaternary contradiction between the object of the overarching activity system and the tools of the 'overall user activity system' – is presented in Figure 7.12.

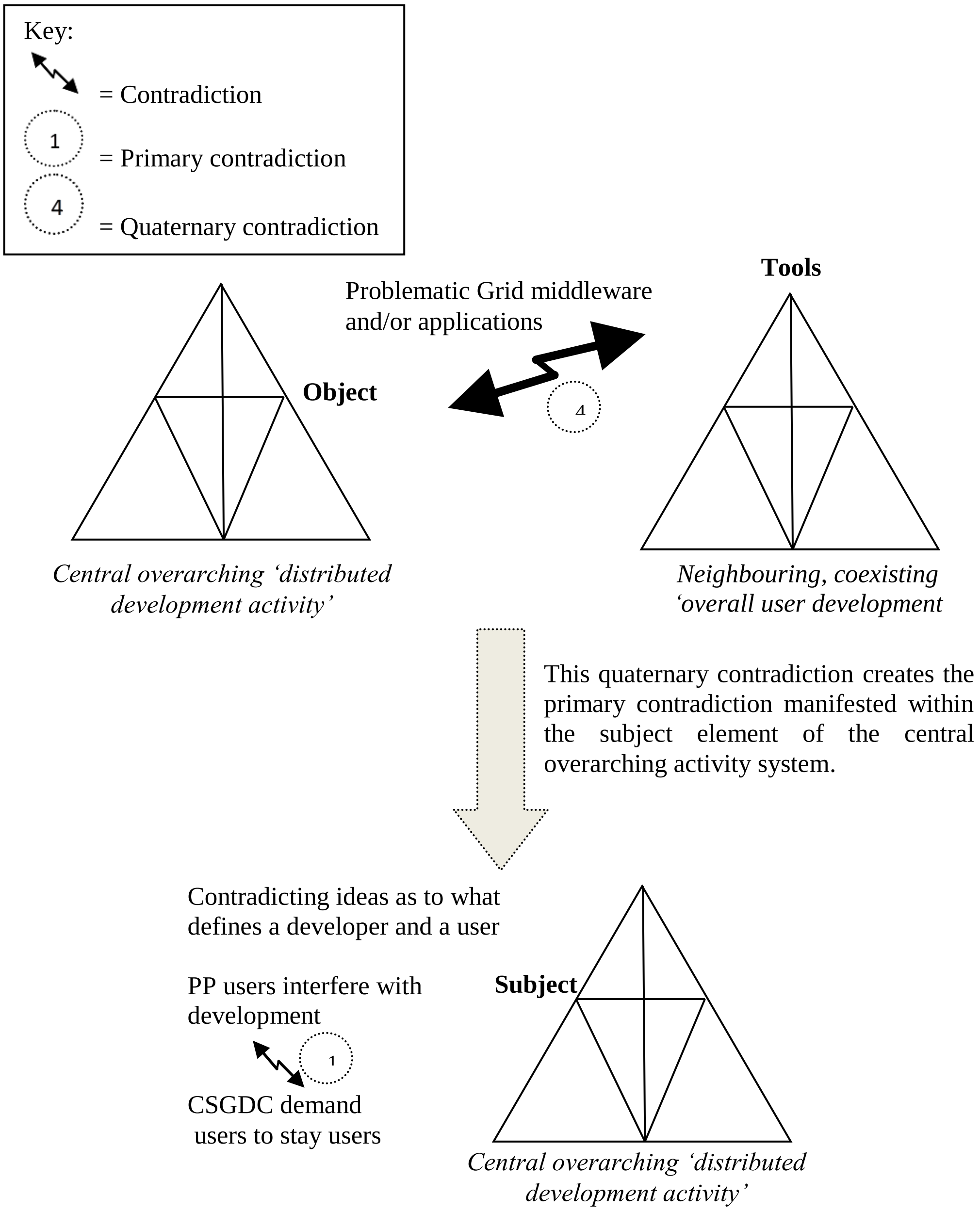


Figure 7.12 Primary contradiction within the subject element of the overarching 'distributed systems development' activity system – Manifested because of a quaternary contradiction between the overarching activity system and the neighbouring 'overall user activity system'

Figure 7.13 presents all the contradictions revealed within and between the elements of the central overarching 'distributed systems development' activity system.

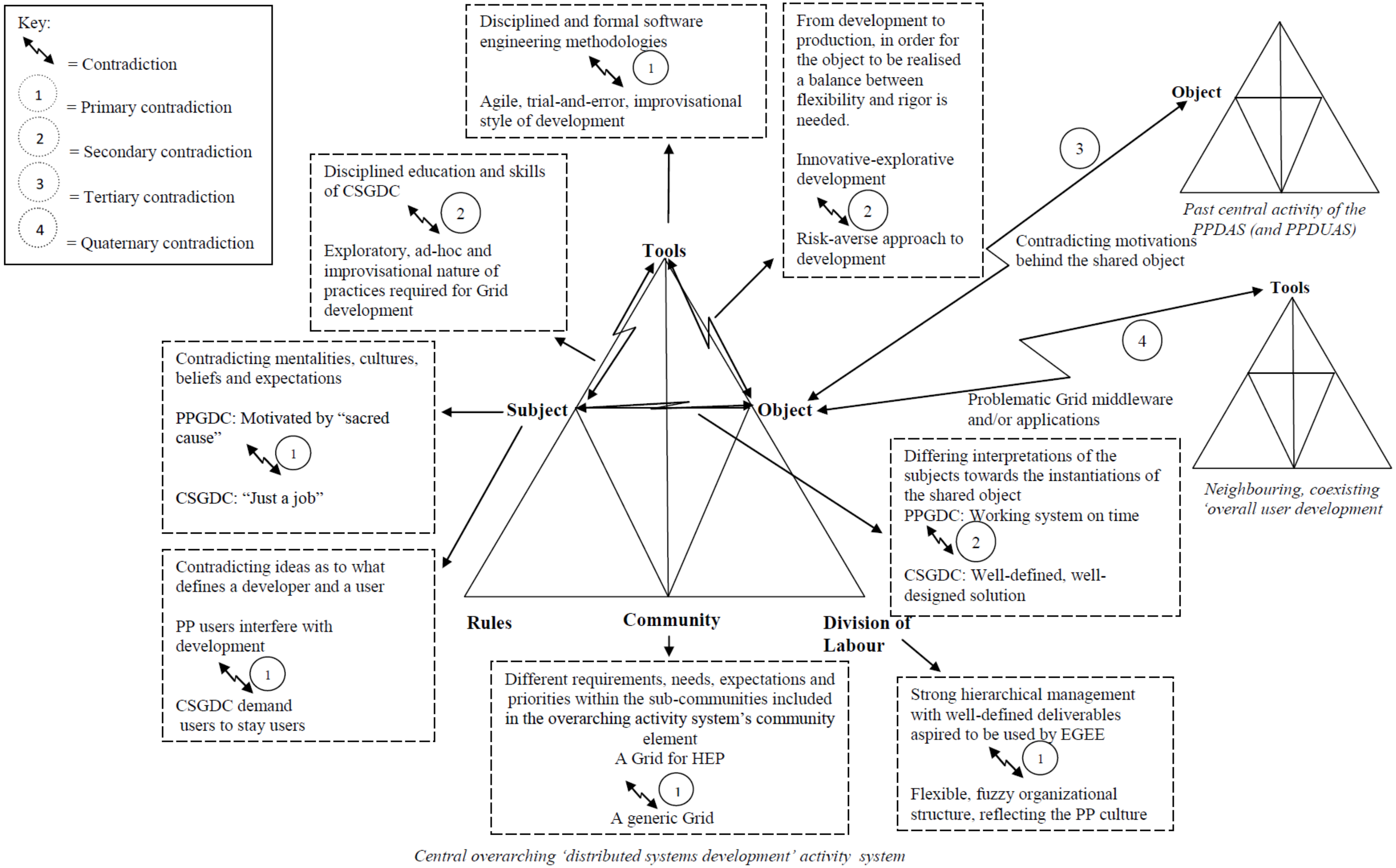


Figure 7.13 Contradictions within and between the elements of the central 'distributed systems development' activity system

### 7.3 Resolution of contradictions through expansive learning cycles

In the previous section a number of contradictions were identified throughout the development process. The most important ones are i) tensions between LCG and EGEE, ii) tensions between PPGDC and CSGDC within the developers group, iii) tensions between CSGDC and PP users acting as developers, iv) contradictions between traditional models to development and current practice, and finally v) problems presented because of the highly distributed environment and the necessity to somehow structure and provide more discipline to the development work.

PPGDC acknowledge that the distributed development of the Grid is a challenging learning journey. Such a distributed development activity is not a homogeneous entity, rather it comprises a variety of disparate elements, viewpoints and expectations (Engestrom 1987). The subjects' means of resolving these contradictions provide evidence of their "expansive learning" (Engestrom 1999c). Such learning drives the progress of the development and not only gives rise to technological innovations (such as the Grid middleware) and development tools to support their distributed collaboration but also to new work practices which enable developers to better face the demands of such large-scale distributed development. Through this expansive learning, therefore, a shared collective mind is developed over time, which enables people to improve their community's capacity for collective action (Weick and Roberts 1993). These emergent practices are now discussed in detail, focusing on the problems faced and how these are resolved.

#### 7.3.1 Divergent motives, interests and expectations: Contradiction resolution

Different people, depending on their role in the projects of the GDC seem to be either oriented towards the ultimate object, or more oriented towards the slight concrete goals that need to be undertaken in order for the common object to be achieved. Occasionally, however, the individual efforts get sidetracked into things that may not be useful for the whole PP collaboration, and the different groups' actions do not necessarily align to one goal (Leont'ev 1981). This primarily happens because of the differing expectations regarding the final Grid, the divergent priorities and motivations and the clashing of cultures and mentalities which make the spontaneous actions at an individual level seem chaotic, undirected and away from the common goal. As a physicist developer argued: "*We just care about a working* *system. Other people for example care about their funding and so this is where you get the different motivations and the goals going in different directions. And so you need to find a way of trying to make the goals mutual".*

Despite these divergent actions, the day-to-day development activity is not undirected and ultimately they all work towards the shared objective. The coordination of the development work drawing upon the PP experimental culture, intellectual arrogance and history of successes, seems to ensure that such seemingly extemporaneous actions add-up despite the complexity and uncertainty of the project and the divergent motivations and expectations: *"You have to talk, talk, talk and find a compromise, because there is nobody who really can impose something on somebody else. The funding lines are different, the needs or the goals are different. So the important thing is that you unite everybody on a common baseline and say this is the objective, this is what we want to do and this common baseline at the moment is what we want in order to build a computing infrastructure"*.

The PP projects' minimal but sufficient structures and plans, the charismatic leadership, the context of collaboration, the consensus-based decision-making, the high-level shared vision of "doing physics" and the continuous and extensive communication flows orientate and somehow direct the otherwise "drifting" (Ciborra and Associates 2000) development activity to make it align with the shared object. The clearly articulated goals provide momentum and a sense of direction in terms of objectives and shared vision (Swanson and Ramiller 1997), which operate through culture without prescribing individual action, but rather normatively shaping such action, while on the other hand the short-term milestones build up a sense of urgency and therefore help to keep track of the variations between disperse innovative actions and define priorities within the collective goal. In other words, although the daily actions may be unplanned, ad hoc and drifting (Ciborra 2002), the minimal strategic planning and flexible management ensure that this "drifting" is appropriately oriented towards a clearly articulated goal and when necessary can perform mutual adjustment between the goal and improvisations. It is indeed argued that: *"The people who are driving the project try to create a specific interest or a specific momentum around the common goal, so although there are a lot of scatters here and there, going in different directions, at the end the goal is the same"*.

Nevertheless, not all efforts get aligned. Tensions between CSGDC and PP users, stemming from their contradicting goals and mentalities, are certainly present and there is no way to eliminate them. The long tradition of computing and a history of technological successes within the PP community empowers PP users who feel that they can interfere with the development work, break interfaces, develop parallel solutions and submit jobs only to the good sites, blacklisting the rest (Venters, Zheng et al. 2007). The subjects involved in the overarching 'distributed systems development' activity system therefore cannot commonly agree to one solution, or refuse to realise that problems exist, resulting in this ongoing contradiction to continue to exist (Virkkunen and Kuutti 2000). There is, however, enough commonality within the project to keep all efforts going and it is indeed believed that this contradicting situation of diverging goals is good for competition and ensures that the most relevant technological ideas survive: *"You have diverging goals and different interests which is, on the one hand, good for the competition as you ensure relevant solutions. But, on the other hand, there is enough commonality, like in a tree structure, where you have the common trunk and then you have the different leaves and parts of the tree going everywhere because you explore the whole space".*

The next sections identify the collaborative practices that emerge from the resolution of contradictions.

### 7.3.2 Expansive learning cycles – Lessons learned

The resolution of inner contradictions proceeds through cycles of emergence, transformation and resolution. Subjects involved in the 'distributed systems development' activity in their attempt to remove these disturbances, alter and develop further the cultural mediators of their activity (Kaptelinin and Nardi 2006). It is in this transformation of their organisational collaborative practices that valuable lessons for those engaged in distributed systems development are found.

Throughout this section the expansive learning cycle diagrams are used. While these diagrams may suggest a cyclical activity, as recursive loops may occur at any point between and within the single steps of expansive learning, they are here used to demonstrate an instance of an expansive learning cycle which occurred through the resolution of a number of inner contradictions for a specific period of time. In particular, the various diagrams demonstrate learning cycles which took different lengths of time and they therefore emphasize the historical development of the community's practice. The end point of the cycles was necessarily, however, the time of the completion of the data collection phase of this research. These diagrams may appear somewhat rigid in their form, however this representation is only for the sake of simplicity and further expansive learning cycles may follow or produced.

**1) Establishment of a development process**

After almost 10 years of performing distributed Grid development the GDC's projects have matured and gained enough crucial experience in terms of doing development work. Developing the Grid has indeed been a learning journey, where both success and failure were inevitable, experimentation was certainly present and ad-hoc solutions were sometimes found to outweigh the well-designed technical systems. However, in the early days, development was chaotic, drifting and the software developed could not live up to the standards expected by users. Incompatibility issues, bugs, difficult deployment and minimal user support were all factors hindering the work causing tensions between the subjects involved in the overarching 'distributed systems development' activity system who questioned the existing standard practice (Engestrom 2003).

This dissatisfaction, leading to stress and conflicts, presented a contradiction and made the collaborators realise that in order for such a contradiction to be resolved, there was a need for modelling a new framework (Engestrom 1999a), which would provide more control and structure in the development work. EGEE, in their attempt to provide some structure, decided to implement and employ a "proper" software development process where in order for any software to get into production it had to pass through significant formal testing, from the PPS and to be accepted as a production system.

The need for more rigorous testing made the software development cycle fairly established. However, the process is found to be constraining, inflexible and ineffective by PPGDC since, as they argue, it takes too long for the system to go into production. This contradicts the old way of working within the activity system and is untypical for HEP activities: *"The process as it is done now allows us to find serious* *problems only in production...the feedback that comes at each stage is very slow…the testing procedure is ineffective…it is impossible to test software at a production scale except by putting it into production. So problems come to light when they actually go into production".* Therefore, scalability problems, reliability problems, and stability problems exist, usually due to high load. Adopting this new process thus creates a further contradiction between the old practices and the new ones, since the process is not followed religiously by PPGDC who prefer *"an internal software process that allows them to be very flexible and dynamic".* A constant tension between this expansive oriented solution and the old one (Engestrom 2001) is certainly present with PPGDC trying to find ways to return to their old practices. Indeed an EGEE leader indicated: *"What we are trying to do is put a bit of method in the madness…and loads of people are outraged – oh, this doesn't work, it reduces my freedom and so on".*

PPGDC subjects involved in the development activity and even some CSGDC have established informal mechanisms that allow them to bypass the EGEE's formal process, so resolving this further contradiction to them. As a middleware developer explained: *"We shortcut the old procedure by introducing what are called the experimental services. So we can have a very fast feedback and we also are allowed to do very quick changes. This proved very effective. So we tried to adjust the formal software process with this experimental service".* Different developer groups within the subjects of the central activity system use their own bypassing procedures, but also different sites have established their informal PPS, after the actual production of the software in order to ensure user satisfaction. This, however, results in standardisation issues and duplication of efforts giving rise to new problems and breakdowns.

The realisation of the ineffectiveness of the already established process made EGEE attempt to provide a solution to this problem, which gradually gave way to a new practice, which is quite different from the already planned model of the new activity (Virkkunen and Kuutti 2000). The new practice of the activity "formalises" this bypassing in EGEE's attempt to satisfy both the developer and user communities. The new process has now been put in place but it is still to be seen whether it will be successful or if it will continue to get bypassed, leading to further expansive learning cycles. First impressions from EGEE's perspective however show that: *"It is not very successful because, at the end, it is more work*". A new model of the activity is therefore now created which is more oriented towards immediate feedback: *"So this fast feedback is very important and this is the direction we are going in more and more in the future.*

The expansive learning process of establishing a software development process is presented in Figure 7.14.

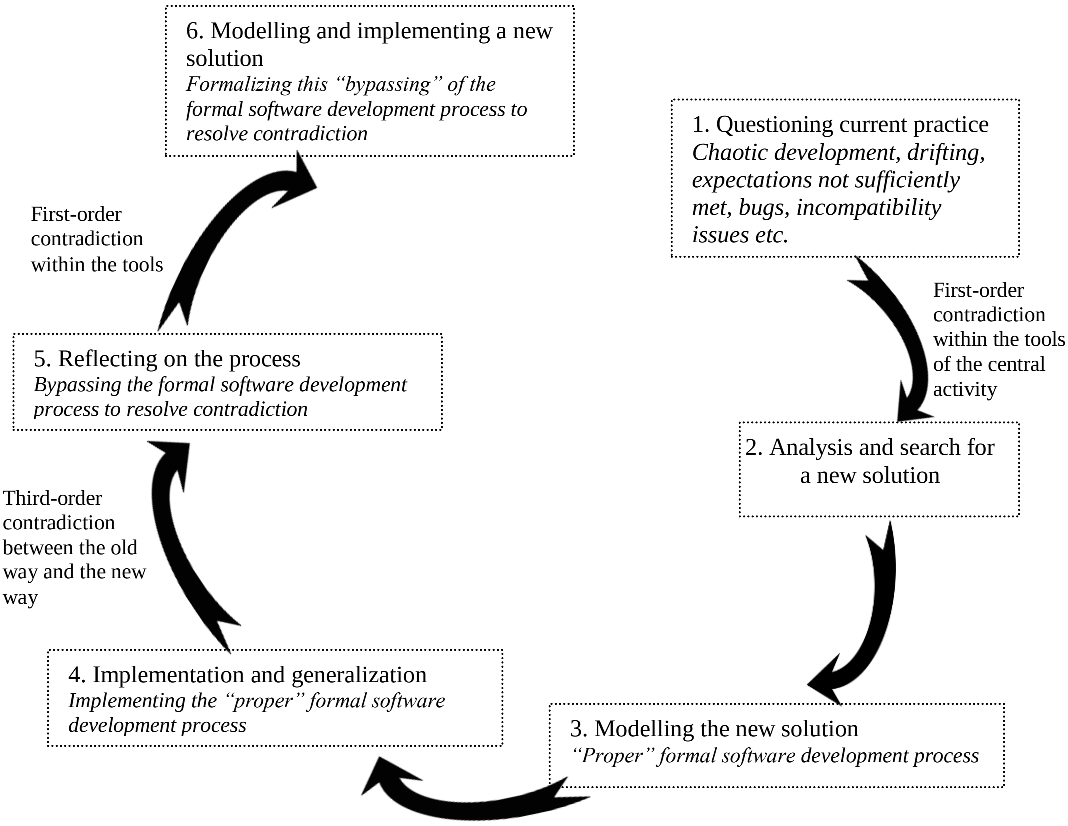


Figure 7.14 Establishment of a development process: Expansive learning cycle with types of contradictions

**2) Clusters of competence**

One major problem faced during early middleware development was the difficulty of getting different distributed groups of people with different backgrounds working on the same component to work together, collaborate and develop at the same speed and using the same approaches. PPGDC indeed argue that: *"We have tried distributed development in the past and it is really impossible...For our types of projects where a strict specification cannot be developed upfront as nobody knows what they final* *system will look like exactly and we have moving targets for many things, then fully distributed development is chaotic".*

Distributed development was found to be a painful process, creating communication overheads and leading to disappointing outcomes *"as middleware components could not fit together"*. Delivery of components was slow, developers could not easily react to problems promptly, thus making it difficult to develop quick prototypes and get immediate feedback. The different projects of the GDC had generally found that distributed development was inefficient in terms of coordination, managing the dependencies, sharing of knowledge and handling funding for such a large-scale experimental project. Getting things moving was hard, but it was even harder to create a shared motivation, something crucial for sustaining the collaboration, eliminating conflicts and therefore maintaining the momentum in the project.

Their answer to this contradiction was to model and structure the development work into "clusters of competence". This is the way in which particular groups become experts in a particular area of the Grid and co-locate (or emerge in one physical location) to reduce the difficulties of needing to constantly communicate and collaborate with others undertaking similar work*: "We are trying to get to a situation where one component is being developed by the same group of people who are all in one place. You try to put some structure in the distributed environment and you try to get people doing the same thing together".* The idea of the 'clusters of competence', as a senior LCG team leader explained, "*means that you can only be a player if you have either enough people who are one hundred percent dedicated, so you become a cluster of competence, or that you already have people who have a significant amount of experience and then you can play with a few new people that you add, and so basically the training has to be provided invisibly by the cluster, because local training is much more efficient".* They therefore created different globally distributed patches of expertise, where experts are co-located, but are then all aligned into a network that facilitates and coordinates the work and the collaboration. As a particle physicist developer explained: *"We have come up with the idea of strong collaboration between the developers in order to manage the dependencies. Now all the dependencies are such, that there are no conflicts"*. The solution to this contradiction aspired to enable better collaboration, minimise competition and achieve a continuity of collaboration between the distributed clusters in order to create and sustain a shared vision guiding their work.

Such networked clusters of competence facilitate communication and sharing of knowledge and expertise among the different clusters: *"The idea here is to have strong collaboration within the developers group as well as collaboration with other countries in order to share knowledge and expertise".* Strong collaboration between the different clusters is required and this is because: *"The piece of software developed has to be used in a bigger piece, in a big environment in a bigger architecture and therefore the pieces eventually need to fit together in order to form something homogenous and for that, collaboration is essential".*

It is indeed stressed that the mistakes previously made especially during the first years of Grid development, such as performing distributed development with minimal collaboration should be avoided. As an LCG senior member explained: *"There was no way of making the software work because developers were making different assumptions on the libraries, on the interfaces of the components, on everything...so that mistake had to be avoided".* Collaboration is needed in order to ensure that the different components are compatible and can work together and run common interfaces.

Although the clusters of competence are found to provide discipline in messy situations, for the development of some components of the Grid this is not possible; therefore this new way of structuring development work in some cases contradicts the old ways of working (Engestrom 1999c). Components being developed in a truly distributed way still exist, resulting in the old problems reappearing. Their attempt to alleviate this further contradiction is not however to return to their old unruly practice of doing distributed development (Virkkunen and Kuutti 2000). Rather, they proposed a new intermediate practice where although parts of the component would be developed in several places, the coordination of the whole component would be performed from a single location. In this way communication, control and sharing of knowledge would be made easier. In order to facilitate this new way of acting and so make it more effective they supported it further by encouraging temporary co-location of developers for a few months and therefore establishing frequent check points to enhance collaborative relationships, resolve problems, improve awareness, forge an understanding and enable strategic thinking: *"Developers from other countries come here and work for 3 months and this is much more efficient…It's much easier to control the activity rather than when people are away. There is still freedom and a level of autonomy but for large features we discuss and decide how to proceed. So there is freedom in a controlled way"*.

The expansive learning process of structuring the development work into "clusters of competence" is presented in Figure 7.15.

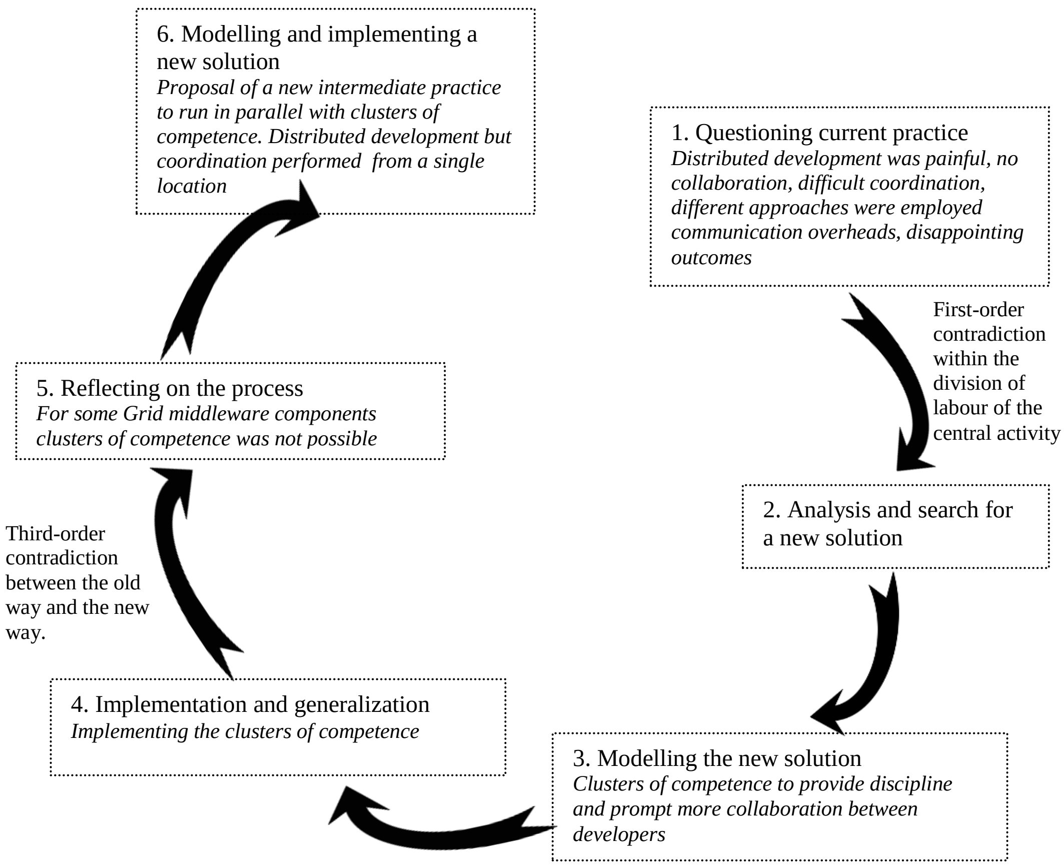


Figure 7.15 Clusters of competence: Expansive learning cycle with types of contradictions

**3) Cultivation of power user communities: A strong sense of belonging**

One problem faced throughout the development and deployment process was the communication breakdowns between developers (mostly CSGDC) and PP users. Although a lot of money was going into support, the relation between CSGDC and users still seemed to be decoupled. Collaborative tools such as GGUS were not perceived as useful by PP users as it was argued that they were not supported by people who *"understood what they were supporting"*, with most users not using them, becoming frustrated, and rejecting to use the Grid. Being powerful agents, PP users sought alternative ways to do their work, which sometimes meant duplication of solutions and wasting of efforts.

This contradiction presented a serious problem, as users would refuse to use the Grid. This reluctance resulted from the lack of proper communication and the lack of a sort of structure directing the work. Although EGEE's response was to attempt to improve GGUS and other technical tools to facilitate user support, LCG's response was the cultivation of the so-called 'power users' who are local people being considered experts of both the experiment and the Grid and who can help facilitate users with their daily problems in order to allow the developers to do development work. These 'power users' would carry the knowledge forward through being observed by the next power user and through socialisation. Although 'power users' was a quite interesting approach to the problem, what was observed here, and it is worth noting, is the cultivation of power user communities (the GridPP storage community is taken as an example here) which is not something pre-specified or pre-designed by the management, rather it is a spontaneous act of a virtual distributed group creation, managed largely bottom up. It emerged because of the need to support users dealing with the storage system and while in the beginning there was only one person funded to really manage and support all storage stuff, a virtual community, reflecting some of the features of a CoP (Wenger 1998), was created around that effort with the aim, as a member explained, to *"foster this idea where everyone collaborates together on this storage issue, and a community of sites is created who can help each other out"*. The role of this community was not just to provide support, but also to build up links with the sites, end-users and the developers thereby creating a *"community spirit"* where everybody can collaborate and exchange ideas and thoughts. This feeling of creating a sense of belonging to a community traces back to the PP tradition and it indeed provides evidence of the expansive learning of the people involved in this community (Engestrom, Engestrom et al. 2002).

Interestingly, when PPGDC were asked what other communities could learn from the distributed development of their Grid, interviewees suggested that what makes their project progress is a combination of factors. As they argued: *"We've many times* *seen the development of systems by isolated groups involving formal procedures. But these didn't have good results".* Therefore something more than co-location and formal procedures is needed in order for such kind of virtual projects to be successful: *"Social is the key really. It makes such a huge difference when people work together for the right reason. True quality comes from within"*.

Creating a strong sense of community with shared goals is found to be crucial for their collaboration and is something they learned through years of experience in working in a distributed manner: *"So it is not so much a software development, the story we have to tell, it is building this community around the computing Grid... Collaboration and building a sense of community is really important for distributed development. We work a lot using mailing lists; you can see the different attitude people have before and after they meet in person through those mailing lists".*

Shared goals provide motivation and an identity is constructed for those involved in the development of the Grid. This sense of community is related to the frequency of the face-to-face interactions, the extensive communication flows, timely feedback, keeping all people involved, creating a trustworthy environment, but also depends on the equally important indicators of shared identity such as logoed pens, posters, T-shirts worn at conferences, etc., and an intense focus on disseminating the project's successes by discussing their work on every occasion.

The feeling of belonging to a group also balances competitive relationships and as they argue: *"Proper management of competition leads to successful outcomes".* Although they might have competition throughout the process, in the end there is a need for collaboration. It is argued that the reason the LCG project is considered successful is because *"it has pulled its community, or the community has pulled around together and acted as a coherent whole to meet its goals".*

This 'power user' community, therefore, being an example of such a collaborative attitude by creating and facilitating a strong sense of belonging between its members, emerged, facilitating the development of a collective mind, which was enacted when individuals engaged in this activity and constructed mutually shared fields to respond to the contradiction (Weick and Roberts 1993). This community did not attempt to bring in some structure as such, rather to make sure that experiences are shared and to minimise duplication of efforts. As a member of this community argued: *"Coordination and communication is really the primary goal of the group and then the structure and the management and planning is really secondary. Less effort is put in when people share experiences".*

While a lot of effort is put into getting all sites and developers involved, this "emerging community solution" to the contradiction proved to be successful and it is now fairly established, leading other such communities to emerge. The GridPP storage community is expanding, reaching a total of more than 60 members. Through time the community has become a credible available source of information, experience and knowledge and it is highly valued both inside and outside the UK.

The expansive learning process that leads to the emergence of the power user community is presented in Figure 7.16.

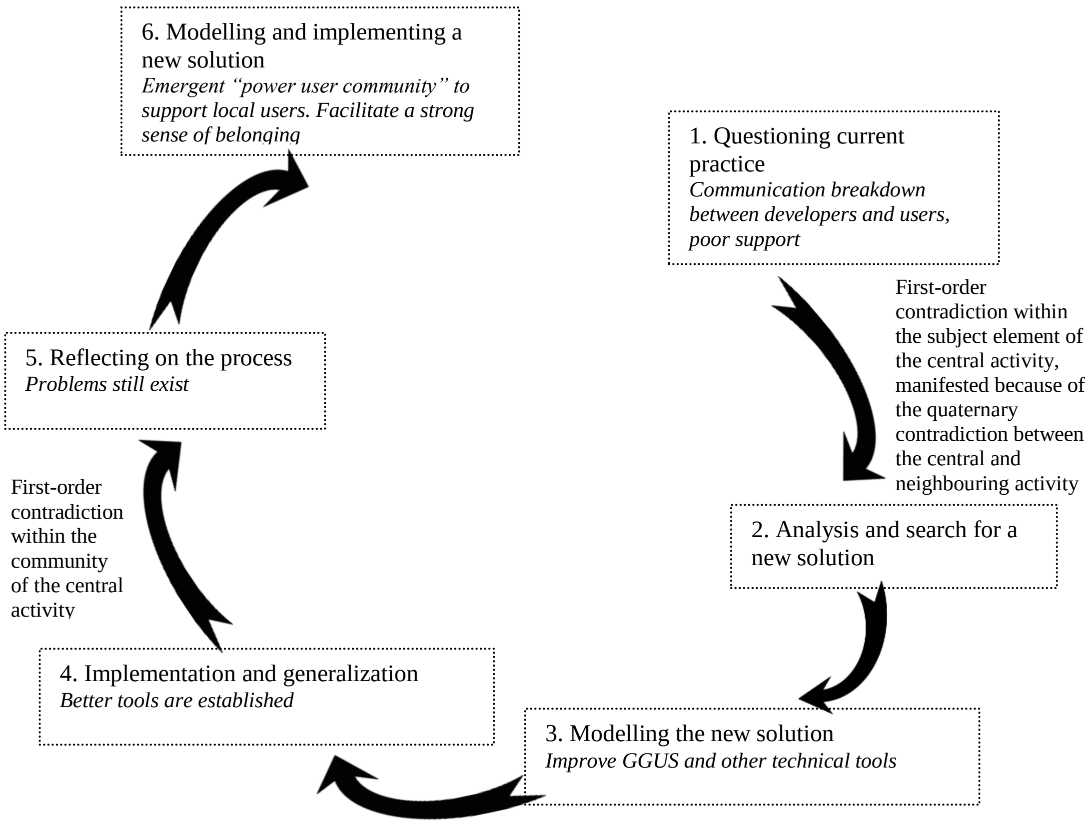


Figure 7.16 Cultivating power user communities: Expansive learning cycle with types of contradictions

**4) Balancing experimentation with discipline**

All PPGDC must write computer software in order to undertake their physics analysis since packaged applications for this task do not exist. Although they have a tradition in computing, they do not have formal training in software engineering or traditional systems development: *"As a physicist you do not get much experience in writing software which "stays up" and is reliable. We work through trial and error and through this you do not get the experience in writing code for stable services".* They are pragmatic, "dirty programmers" who like working solutions. They believe in producing things that work quickly and that is why they do not usually prefer the "fancy" well-thought and well-designed solutions that take time to be developed. Indeed one of the interviewees, jokingly said: *"We are intelligent people; we don't make bugs so there is no need for methodologies".* For them, developing software is an experimental activity involving trial-and-error in a way similar to the way physics itself is undertaken. They are not constructing a pure system, they experiment and therefore if something does not work, it does not matter to them, they will just do it in another way. However, this mentality in developing the Grid frustrates CSGDC who sometimes worry about the delivery of a system which might stay up and running for a few years but will not be able to provide a reliable service to all the user communities, not just physics.

When asked about this unstructured "experimental" (and arguably risky) way of working, most PPGDC agreed that such distributed development projects must combine this agility/flexibility together with limited structure/discipline: *"One thing the project learned is that you need management and clear short-term priorities, or else you drift"*. Because the GDC's projects consist of both PPGDC and CSGDC, some discipline in development activities is seen as necessary to balance the developers' individual goals with the shared object, but also to keep the project on the right track, submitting by deadlines and within budget. It is also crucial, however, to maintain this flexible and agile character in the way they work in order to quickly adapt and respond to environmental changes and quickly develop prototypes to get quick feedback.

Their response to this contradiction, especially as they move from a development stage to a production stage is to use collaborative processes that provide discipline and control over the development, but also allow for “shortcuts” and quick modifications. Indeed a middleware developer stated: *“You have to be far more agile with the trends in software engineering and programming…And here, you do have to do that. And I don't think it is a bad thing. It doesn't mean that you should drop the methods, it just means you should rotate them faster, and have small steps, etc. And this is what we are trying to do, and when we are successful in doing it, in some parts, the project goes better, for sure. Every time we stop and try to make bigger changes in one go it really causes problems and people are not happy with that”.*

Stronger management mechanisms have therefore been put in place to facilitate such a change in the way work is performed; however, these again are influenced by the PP tradition exhibiting a culture of “charismatic leadership” and “soft management/leadership”. For example, one form of management in order to make sites perform better is the so-called “Steve’s jobs”. A set of jobs run every night on the Grid thus challenging all sites in order for different problems to be identified. The problematic sites identified are then published online and therefore although there is not a notion of imposing sites to fix the problems, still their need for higher appreciation and recognition of being a successful site makes them resolve the problems.

The expansive learning process that leads to enforcing stronger management mechanisms, ones however reflecting the PP culture is presented in Figure 7.17.

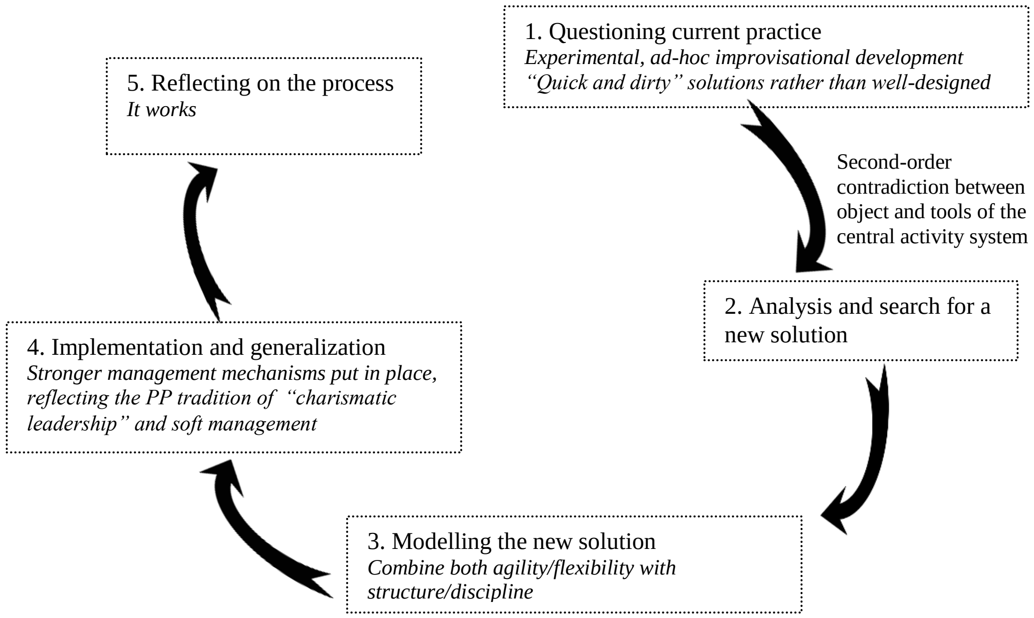


Figure 7.17 Balancing experimentation with discipline: Expansive learning cycle with types of contradictions

## 7.4 Chapter summary

The analysis of the collaborative central overarching ‘distributed systems development’ activity presented in this chapter is two-fold. Firstly, it explicates the intricacies and contradictions between and within the elements of the central overarching activity system faced throughout the development of the Grid and caused by the contradictions emerging from the interlinked networked sub-activity systems comprising the central overarching activity. Secondly, it focuses on how these were resolved in order for the overarching activity system to achieve stability and balance. Contradictions are considered to be the major source of dynamism and development in AT (Nardi 1996), since based on the emerging problems and conflicts, people have to re-consider their position and collectively reconstruct their shared understanding, knowledge and practices. The resolution of such contradictions therefore provided evidence of the community’s expansive learning, leading to new collaborative practices emerging to deal with problems and unexpected opportunities (Engestrom 2000b).

The key questions driving the rationale of this chapter were explored and the following key issues have been identified:

*1) What is/are the activity system(s) involved in the distributed systems development of the Grid?*

The distributed development activity of the Grid is a complex overarching activity system comprising instances of a network of functionally linked sub-activity systems. The first sub-activity system is the PPDAS, the second is the CSDAS and the third is the PPUDAS. The overarching central ‘distributed systems development’ activity system therefore presents a merge of these three sub-activity systems and demonstrates the dominant institutional practice – the PP culture, mentality and style of development. The central overarching activity system has been examined providing information on the elements comprising the activity and how these shaped each other, the object and the desired outcome.

*2) What are the different points of view, interests and goals in relation to the artefact and the development activity?*

A number of contradictions have been revealed within the development activity. The most profound were between the two big projects LCG and EGEE, between CSGDC and PPGDC involved in the development and between CSGDC and PP users acting as developers. These contradictions were mostly in terms of conflicting motivations, differing expectations regarding the Grid artefact, differing styles of development and the power of PP users as active agents influencing and shaping the development work and outcomes.

3) *How are these contradictions resolved in order for the activity system to achieve stability, facilitating learning and the evolution of systems development practices?*
The resolution of these contradictions necessitated extensive communications, socialisation and consensual negotiations and sometimes not all contradictions could be resolved either because subjects could not commonly agree to one solution, or because they refused to realise that there were problems present (Virkkunen and Kuutti 2000).

Resolving contradictions facilitated learning and therefore a number of strategies for distributed development can be extracted based on the lessons particle physicists learnt throughout the Grid's development. While particle physicists' collaboration has been described as "exceptional" (Chompalov, Genuth et al. 2002), their means of coping appear both unusual and yet somewhat orthodox. In summary, the strategies identified were: 1) To establish a development process but one which allows for short cuts, and is very flexible and dynamic, 2) To structure the development effort in clusters of competence, 3) To encourage temporary co-location of developers, 4) To cultivate 'power user' communities, 5) To combine flexibility/agility with structure/discipline, 6) To create a sense of belonging and therefore construct identity for those involved in the development, 7) To facilitate human communication both through virtual means and face-to-face (at least every couple of months), 8) To create a trustworthy environment, 9) To have clear shared goals and rationale.

The next chapter draws together the literature (presented in Chapter 2) and the analysis undertaken in this chapter in order to present a thorough discussion of the findings of this study. A framework of guidance for GSD is developed and practical recommendations are provided.

# 8 Discussion of Hybrid Experimental Agile Distributed Systems Development Communities

The analysis of the case in the previous chapter demonstrated the contradictions that characterise the GDC's distributed systems development activity and how these are resolved leading to lessons which have the potential to be useful for answering the problems faced today by various GSD projects.

In this chapter, reflecting on the case material, on the analysis and through the lenses of various literature (Chapter 2) and AT (Chapter 3), practical implications for the wider discourse of GSD and agile development are drawn and a framework of HEAD_SDC is developed and provided for those engaged in engendering similar distributed systems development practices. While GSD projects are apparently directed towards enhancing the strategic flexibility of organisations, it is argued that in many aspects they are still quite rigid (Breu and Hemingway 2004). In an age that demands "faster-cycle technological innovation" it is important for such projects to develop agility (Lui and Piccoli 2007). However, existing ISD methods, such as agile methods have been proved to be inflexible for these projects with authors suggesting that other potent bases of agility for distributed projects need to be investigated (Lee, Delone et al. 2006; Conboy 2009). Large-scale global systems development thus presents unaddressed challenges to organisations and it is argued that there is an urgent need for new systems development practices and strategies that can facilitate and embrace the rapid changes of the environment, and the complexities involved in such projects (Herbsleb 2007).

The aim of this chapter is, therefore, to provide answers to these problems and to fill in the "gap" identified in the GSD literature through the lessons learned from the distributed development of the PP Grid. It particularly reveals a new set of fluid dynamic collaborative practices that can inform the ISD literature and practice. The framework developed here does not provide a linear process for guiding the Grids' or other large-scale systems' distributed development process. Rather, it is a structured set of concepts and ideas that emerge from this study. In particular, it provides a structured discussion about the various things (technical, organisational and social), which the GDC needed in order to "achieve" their Grid. Although particle physicists' work is unusual and somewhat unorthodox this framework may allow other organisational contexts to reflect on their own practice and perhaps foster such a similarly dynamic, flexible and agile community in order to manage their GSD efforts.

The chapter begins by examining the framework's ideas and comparing them to already existing literature on communities (discussed in Chapter 2). This is followed by an in-depth description of the most important characteristics of such HEAD_SDC focusing on the structure, history/culture and processes of such community (Section 8.1). The relevance of this framework to GSD and agile development literature is portrayed, after which the framework is presented (Section 8.2). Section 8.3 summarises the chapter.

## 8.1 HEAD_SDC: Structure, history/culture and process

This study suggests that the GDC explored in this study represents what the researcher defines as a HEAD_SDC, which reflects some of the elements of what Porra (1999) describes as colonial systems and exhibits similar but also additional and distinctive characteristics to the ones already defined in a traditional CoP (Wenger 1998). Developing further the literature on CoP and colonies, this community is identified as a new form of collaboration, which is highly experimental and analytical, with significant computing expertise and not following predetermined plans, methods and structures. This community is nurtured and sustained through the years by the whole collaboration of particle physicists, rather than spontaneously emerging as a traditional CoP (Huang, Newell et al. 2003; Schwen and Hara 2003). It is bounded by socially constructed rules and ethics that promote the formation of shared ideologies and cultural forms and its collective evolutionary history plays a vital role in its survival, like in a colony (Porra 1999).

While a CoP is a self-directed and self-motivated entity and the engine that drives it is the shared interests of its members, which may not be the same as those of the wider organisation, the HEAD_SDC has one general shared object and all its members, despite their sometimes differing and contradicting interests, need to work together in order to accomplish it.

Such a HEAD_SDC has unique characteristics because of its unique history and culture, which are widely shared between its members through socialisation, sharing stories, etc. The culture of the collaboration in this community passes on from generation to generation as customs, norms, values, stories and behaviour and, (as argued by Porra and Parks (2006) for a colony), these are designed into the community as rules of conduct. Members of such a HEAD_SDC engage in the same kind of work, identify with this work, share norms, values and perspectives and their social relationships meld the realms of work and leisure thus creating strong bonds between them without having a single centre of supreme skill and authority like in a CoP (Engestrom 2007). Unlike formal organisations which are more strongly based on career and monetary contracts, this community is based on seemingly spontaneous acts for achieving a "sacred cause", has the ability to evolve and is founded on bindings such as long-lasting common interests and shared identity. It is directed and motivated by shared visions and while it may appear leaderless without clear authorities for goal settings, power or control, yet, no other outside entity can impose its authority over the community without its consent (Porra 1999).

This community, unlike CoP, (Kimble and Hildreth 2004) is highly distributed and virtual and its computing expertise in dealing with online collaborative technologies ensures learning and skills development, although face-to-face interaction is still valued. In contrast to CoP or colonies, such HEAD_SDC's collaboration is clustered and consists of a larger, loosely knit, geographically distributed group of individuals engaged in a shared practice similar to NoP (Brown and Duguid 2001). However, and in contrast to NoP, people are organised as a collective rather than at a more individual level and invest a lot in socialisation. Coordination of the different clusters as well as knowledge flow is provided from the communal social network connections between the clusters rather than through third parties or only through conferences as defined by Brown and Duguid (2001). Building a strong community bond between the clusters is important and therefore such a community facilitates frequent face-to-face occurrences.

It is here observed that members of such HEAD_SDC are knowledgeable actors (Orlikowski 2002) and their knowledgeability is continually enacted through their everyday activity and through their improvisation and spontaneous agile actions (ibid). The Grid project seems to be constantly at risk and loosely led, the daily practices seem to be unplanned and "drifting" (Ciborra and Associates 2000), yet, the minimal strategic planning and "charismatic leadership" lead to coherence and support mutual sense-making (Weick 1999). The interaction and coordination among the HEAD_SDC's knowledgeable members allows for the creation of a "collective mind" guiding and informing the work and the community's capacity for collective action (Weick, Sutcliffe et al. 1999) at a globally distributed environment. This "collective mind", which usually happens at the community level and not at the individual, increases the "comprehension of complexity and loosens tight coupling" (ibid) and therefore allows for more effective distributed working. Such HEAD_SDC's collective mindfulness at a global scale is maintained and develops further through the creation of a "*shared vision*" and by providing enough "*inspiration*" and "*momentum*", all important characteristics for creating a collective capability in distributed organising to keep the project going (Orlikowski 2002).

It is evident that the concrete shared vision establishes a sense of urgency, builds community bonds and constructs shared identity, fosters individual commitment and devotion and balances competition with collaboration; the inspiration projects charisma and pride, assures joint accountability, unites contested interests and facilitates the creation of a trustworthy environment where knowledge is openly created and shared; while the short-term clearly articulated goals provide momentum and motivate people to keep the project going. Furthermore, it is here observed that such HEAD_SDC's response to the "distributed environment" contradiction is by structuring the development work into *"clusters of competence"*, which ensures the shaping of purposeful, goal-oriented, collaborative and permeable distributed teams consisting of motivated independent thinkers with a natural charisma of voluntarism and autonomy in the way they work, thus ensuring the transmission of the PP culture. This way of structuring development work indeed ensures synergy and coherence between the local clusters as well as allows individuals to know where expertise is located, contributing to the creation of a trustworthy environment.

While such HEAD_SDC is not very stable, as PhD students and post-docs come and go, this study shows that it is still preserved through conversations on how the LHC progresses and as older members of the community share stories of past successful experiences creating a sense of pride of being particle physicists and a strong community bond. It is here suggested that the community bond created contributes to the construction of shared identity and ensures participation and engagement of project members in collaborative work as well as further assures that members feel jointly accountable. The quality of relation or connection between individuals in globally distributed clusters of competence is therefore enhanced by the sharing of stories (Orr 1990), participation in social rituals (Lave and Wenger 1991) and by investing in the relationship.

The HEAD_SDC is now examined in terms of its structure, history/culture and processes. The **structure** represents the division of labour, the coordination mechanisms and the decision-making process of the community. The **history/culture** represent the values, ideologies and norms that drive the community and their crucial role in defining the future activity and its development. Finally the **process** represents the mediating tools (collaborative practices) guiding the work in the community. These three dimensions, identified from the literature (Robey 1991; Nohria 1995; Orlikowski and Yates 2002) and from AT (Engestrom 1987; Nardi 1996), allow for distilling lessons from the PP development activity that can enable a self-reflective analysis by those engaged in similar work.

### 8.1.1 Structure – Division of labour

The identified HEAD_SDC is different from traditional bureaucratic organisations. The structure of this community is *clustered, organic,* highly *distributed* and highly *collaborative.* The work is structured into "clusters of competence" by bringing competence together and strengthening their distributed teams and thus establishing clear links to where expertise and knowledge is located. Organisational boundaries are blurry and not well-defined, with members leaving the community and new members joining at any time. Technical skills, intelligence, self-motivation, independence, "stepping up to do the dirty work" and be familiar with the community's culture, are important characteristics expected for membership. Flexibility in roles, encompassing multiple roles at a time, voluntarism and autonomy in the way the work is being performed, are also characteristics of such a community's division of labour.

The informal organisational structure is "*organic"* as it evolves and changes with time and needs. Indeed, although the LCG project's boards were envisaged from the beginning, their roles evolved to natural niches in the project. Given the lack of formal structure and its fluidity, political power is minimised and the collaboration is managed by *"charismatic leadership"*, rather than by powerful agents imposing authority. Members maintain a general understanding of the project and the shared vision and especially concerning aspects related to their specific roles and daily activities, through the continuous and extensive communication flows in the community. The lack of a formal hierarchy means that coordination is through negotiations and consensual agreements. This decentralised and democratic structure in the division of labour avoids layers of decision-making and bureaucracy and provides incentives for sharing expertise and acquiring knowledge, while encouraging creativity and improvisation.

### 8.1.2 History/culture - Community

This study demonstrates that a HEAD_SDC's history and culture is significant. Most members have a PP background – particularly in the key positions of the project. PP itself has a long history of this type of collaborative work and of using advanced technologies (Knorr-Cetina 1999). The community has always been at the forefront of computing, with examples being the work on the Web (Berners Lee and Fischetti 1997), the use of open-source (Linux) server farms; and more recently the development of the Grid (Doyle 2005). Such a history of computing is important such that together with their history as a discipline are found to strongly influence their practices. Experimental PP is, of course, distinctive. Such history of experiment provides strong justification for the management structures of the project, with few members looking beyond the practices and culture of PP for guidance (despite different expertise existing within the project).

A shared culture therefore emerges from this history and from a tradition of respect for individual creativity and technical expertise. Trust and equality, voluntarism and self-motivation, pragmatic problem solving and 'sacred' shared goals drive the work. The level of commitment and devotion appear high. Members value reputation and seniority and recognition of expertise is important. Maintaining their reputation as good collaborators motivates members of such HEAD_SDC to complete tasks on time and keep the project on track. They believe in shared risks, shared rewards and shared ownership and while individuals have personal career interests at stake, these are somewhat minimised by the "higher cause" for most. This shared culture facilitates a shared understanding on how things work and how work should be performed, as well as facilitates communication and creates a strong sense of belonging and a bond among the participants.

This history and culture therefore define the social structures and knowledge upon which the community is founded. They provide the 'freedom' for experimentation, the structure for innovation and direct the practices of those participating. The strong emotional community bond, the spectacular web of communications, the atmosphere of trust and a culture that appreciates the "aesthetics of imperfection" (Weick 2002) provide individuals and groups with confidence and a safety net to endorse bricolage (Ciborra 2002) and to explore and make mistakes, with the knowledge that failures are legitimate learning experiences, and when managed well, can ultimately contribute to the "sacred cause" of the community. This history and culture forms the keystone for this HEAD_SDC to emerge. The demand for academic publications and "outputs" from the project also leads to a written history being created, and recreated, in each new paper or presentation. These freely available documents allow the history and culture to be visible to new members and other researchers.

### 8.1.3 Process – Mediated tools

The HEAD_SDC responds to the Grid's challenges with a spirit of pragmatism and agility. While constant engagement and negotiations between structured processes and amethodical practices, between flexibility and discipline, between experimentation and rigor and between planning and improvisation exist, the processes employed include heavy doses of experimentation, pragmatism, trial-and-error, improvisation and bricolage (Ciborra 2002).

The development of the Grid is informed by hunches, it relies on ad-hoc solutions and can be seen as an instance of *bricolage*, as it is based on "transforming and reshaping what is already in use, or creatively rearranging components to fulfil new purposes" (Lanzara 1999). Developers improvise in order to make sense of unexpected possibilities and constrains that emerge. Attention and interpretation rather than intention and decision-making drive the development process. Developers revise their sense of what is going on and what can be accomplished through extensive use of prototypes and it is these revised interpretations that guide their action (Weick 1993) and these prototypes which form the fabric of the Grid.

While not following any pre-defined methodologies, members of the HEAD_SDC make substantial efforts to establish a suitable flexible development process; one that allows quick modifications but still provides some discipline. Improvisation within the co-located clusters of competence, therefore, is complimented by some structuring at the distributed level in order to maintain coherency across the project and create a sense of community among participants.

Requirements in this community usually take the form of decisions in informal discussions or threaded messages or discussions on websites, wikis, mailing lists that are available to open review, elaboration and refinement. Experimental bug driven development with frequent incremental releases, a pre-production mini Grid service facilitating testing of components before production and maintenance as evolutionary redevelopment and re-invention are the most important parts of this process, which do not however follow each other in order. This process is more emergent and continuous, more spontaneous, open-ended and more shaped by actions rather than by plans – as seen in most current ISD methods, such as agile and open source methods (Madey, Freeh et al. 2002; Baskerville and Pries-Heje 2004).

This community, however, exhibits agility at a global scale (One-Ki, Probir et al. 2006). While not following any pre-defined agile methods, as these are claimed to be unsuitable for such a globally distributed environment (Ramesh, Cao et al. 2006), it still demonstrates agility by relying on the skills and competences of the HEAD_SDC's knowledgeable members and by emphasising on team empowerment and team accountability. Interestingly, the clustered structure of such HEAD_SDC enables the formation of full skill-complemented teams that are agile, flexible and quickly respond to problems. Such agility is also demonstrated in the daily practice of participants by establishing day-to-day targets, thereby breaking the planning cycle into shorter chunks and building software in smaller iterations. Smaller and frequent releases and continuous integration with concurrent testing also fosters fast feedback and thus allows for faster modifications.

The cultivation of power user communities to support local users and help developers is also a distinctive way of this HEAD_SDC for providing user support. Another distinct feature of this process is the identification and exploitation of technical solutions in the project relying on “natural selection”. The community allows different solutions to compete, until “natural selection” e.g. because of technical failures, lack of funding, etc. defines the one to be followed.

Appendix A provides a table comparing and contrasting agile practices and open source development practices with the PP collaborative development practices identified in this study.

Figure 8.1 summarises the structure, history/culture and process dimension of the HEAD_SDC. Changes in one dimension are likely to be accompanied by concurrent changes in the others. For example, if community members do not value the shared goal (culture), the development work is likely to become chaotic (process) and the clusters of competence (structure) are unlikely to be able to work in a coherent and synergetic way.

<table>
<tr><td rowspan="2">Hybrid<br>Experimental<br>Agile</td><td>Structure (Division of labour)<br>→ Clustered<br>→ Organic<br>→ Decentralised<br>→ Distributed<br>→ Collaborative<br>→ Democratic</td></tr>
<tr><td>History and culture (Community)<br>→ Shared goals<br>→ Computing expertise<br>→ Pragmatism<br>→ Trust</td></tr>
</table>

| | |
|---|---|
| **Distributed Systems Development Community** | → *Strong sense of belonging*<br>→*Voluntarism and Self-motivation*<br>→ *Embraces uncertainty*<br>→ *Facilitates exploration and creativity* |
| | **Process (Mediated tools)**<br>→ *Experimental*<br>→ *Flexible/agile and disciplined*<br>→ *Improvisational*<br>→ *Extensive use of prototypes*<br>→ *Bug driven development*<br>→ *Maintenance as evolutionary redevelopment*<br>→ *Competing technical solutions*<br>→ *Power-user communities* |

Figure 8.1 HEAD_SDC's structure, history/culture and process

The following section develops a framework of guidance towards engendering such a HEAD_SDC, drawing on the analysis of the findings, already reviewed literature and AT.

## 8.2 Developing a framework based on lessons learned from a 'unique' development context

Developing large-scale systems in a distributed manner is a complex activity, one which entails both technical and managerial challenges (Kotlarsky and Van Fenema 2008) that current ISD methods cannot address (Simons 2006) .There is a pressing need, therefore, to develop practices, strategies and alternative perspectives on ISD, that can encompass the amethodical and unplanned nature of distributed systems development as seen in this case study.

The framework developed here provides practical recommendations based on examples of organisational practices from the case that may enable the fostering and nurturing of what this study terms as a HEAD_SDC within organisations that perform GSD. While not believing that such a HEAD_SDC can be prescribed, this study aims to present suggestions for how such a distributed and agile community may be engendered.

Engendering CoP is seen as an innovative way to manage knowledge and sustain innovation (Swan, Scarbrough et al. 2002). It is indeed argued that organisations nowadays show an increased interest because of the possibility to take this old concept to address today's global environment challenges (ibid). While virtual communities, exhibiting characteristics of traditional CoP, are seen as important for organisations that perform development in a global context (Rios, Aguilera et al. 2009), the question of how such communities can be effectively established in a virtual context is still present (Gammelgaard 2010). Empirical insight into how to engender and sustain such innovative virtual communities, which can overcome the "capricious" nature (Wenger 1998) of CoP but can still exhibit characteristics of shared identity, common interests and ones that overcome the issues related to trust, and the sharing of knowledge is thus needed (Gammelgaard 2010).

The HEAD_SDC defined here, however, is more than a traditional CoP, as discussed above; rather it is a community with unique characteristics operating in a highly distributed environment and thus it reflects on the realistic challenges and opportunities of performing GSD in such an environment. Such HEAD_SDC's collaborative practice is based on sense-making, improvisation and bricolage, agility and flexibility. It is about accepting what is unpredictable and uncontrollable, while actively enacting those organisational dimensions and properties of distributed collaboration, such as minimal structure, high-level planning, extensive communication and social bonding, which enhance the capability to be agile, to perform under such uncertain circumstances, to generate coherence and coordinate distributed work. Although the HEAD_SDC retains minimal structures and plans to orient the otherwise drifting development activity, these are not simple structures (as agile methods imply) (Sarker and Sarker 2009); rather the community has complex managerial boards, committees and communication processes, which are necessary to maintain coherence across the distributed clusters of competence and sustain a vital sense of community for independently thinking actors.

Furthermore, while agile methods downplay formalistic project management, documentation, process and contracts (Highsmith 2003), the HEAD_SDC' collaborative practices accommodate these within them, which is vital in a distributed environment, by promoting "charismatic leadership" and "spokespersons" rather than managers and by creating a written history of the work through papers, presentations and wikis. Finally, while agile methods encourage us to keep things simple and lean and not add complexity and sophistication until it is needed (Lero and Conboy 2009), this community demonstrates their nimbleness to respond to unpredictable events by allowing parallel solutions to compete and “natural selection” to decide on the one to follow.

Agile methods probe for constant collaboration and communication in co-located teams, but on a global scale, facilitating such intensive collaborative work is argued to present a challenge (Leffingwell 2007). In this community, however, a collaborative performance is observed through which the distributed Grid emerges in an agile manner. Within such a HEAD_SDC agility is seen as an attribute of their dynamic and improvisational practice and is thus manifested through autonomous, yet collaborative clusters of competence, through knowledge and context sharing, by establishing a culture that embraces uncertainty and allows for explorations, by extensive collaborative and communication processes that facilitate a continuous information flow at all collaboration layers in order to keep members engaged and make them feel ownership and by self-organising power-user communities that are composed by autonomous yet interconnected members spontaneously coming together to support a task.

Agility in this HEAD_SDC, therefore, is about dropping the tools (Weick 1996), “letting go” of control and accepting surprise and risk. Letting go of control does not mean chaos or no organisational structures, rather it means that there is enough trust in people and their skills to allow them to make their choices, and decide what they want to do based on their competence and knowledge. Practitioners who like to achieve some of the attributes of such HEAD_SDC’s performance should, thus, not focus on stimulating communication, control and trust per se, but on supporting and sharing the performance of individuals within the community as well as cultivating a similar structure, culture and process as in the case of this study’s HEAD_SDC. Through this, communication, control, trust and agility can emerge as part of their collaborative practice.

The HEAD_SDC identified in this study is represented by an overarching activity system (as discussed in Chapter 7) comprising a number of smaller networked sub-activity systems that interact, compete and contradict in order to achieve their sub-objectives (Engestrom 2001). Nevertheless, despite contested interests and contradicting relations, the structure (pointing to the division of labour of the overarching activity system), the history and culture (pointing to the community element) and the processes (pointing to the mediated tools of the overarching activity system) are still shared between the sub-activity systems and therefore facilitate a balance and a stability between the overarching activity and its sub-systems, enabling the community to evolve further. This research, therefore, suggests that it is important to be able to identify such an overarching activity system with its sub-activity systems within a global organisation if any attempts at engendering such a HEAD_SDC are to be undertaken. In order for the structure, history/culture and process to be shared within such an overarching activity system (and therefore within the HEAD_SDC) there is a need to foster a similarly dynamic and agile environment like that of the case of this study's HEAD_SDC. Management decisions and actions therefore, have to be fine-tuned towards the unique characteristics of such a hybrid community in order to be able to foster its creation.

The following Table (8.1) provides a diagrammatic presentation of the framework.

| | HEAD_SDC | Transition steps towards HEAD_SDC operating within global environments |
|---|---|---|
| **Structure**<br>**- Division of labour**<br>**- Coordination mechanisms**<br>**- Decision-making process** | ▪ Clustered<br>▪ Organic<br>▪ Decentralised<br>▪ Distributed<br>▪ Collaborative<br>▪ Democratic | 1) Structure the development effort in clusters of competence. "Staff'' the clusters based not only on their set of skills but also on their shared past experiences.<br>2) Accept decentralisation. Drop the tools and let go of control (not chaos).<br>3) Invest in facilitating human communication both through virtual means and face-to-face (at least every couple of months). Face-to-face meetings boost the development.<br>4) Informing and monitoring practices can be used such as virtual daily or weekly meetings to provide a sense of direction.<br>5) Consensus-based decision making. Empower members to make decisions by discussions and voting.<br>6) Embrace informal structures and communication channels. |

| | | |
|---|---|---|
| | | 7) Value your people and employ people with intelligence, self-motivation, autonomy and commitment.<br>8) Establish a fluid hierarchy with limited central lines of authority that values reputation, seniority and expertise. Avoid central commands and control structures.<br>9) Provide mechanisms for members to complete tasks through informal relationships and networking.<br>10) Allow freedom and autonomy in the way roles and responsibilities are acquired but in a controlled way. |
| **History/culture - Values, ideologies and norms of the community** | ▪ Shared goals<br>▪ Computing expertise<br>▪ Pragmatism<br>▪ Trust<br>▪ Strong sense of belonging<br>▪ Voluntarism and self motivation<br>▪ Embraces uncertainty<br>▪ Facilitates exploration and creativity | 1) Have clear, shared goals and rationale. These provide impetus to the project and a strong driving force.<br>2) Allow for mistakes and unsuccessful explorations.<br>3) Embrace uncertainty and risks and allow space for creativity and innovation.<br>4) Invest in socialisation as it is crucial in creating community bonds and shared history.<br>5) Enable free flow of information. Share knowledge, minutes of meetings etc. through wikis, emails etc. in order to keep people engaged and make them feel they belong to the community and are a part of this effort.<br>6) Establish clear links to where expertise and knowledge is located, thereby establishing trust and facilitate learning.<br>7) Create and maintain a sense of belonging and therefore construct identity for those involved. Facilitate the development of a strong bond among collaborators through i) frequent face-to-face and virtual meetings, telephone conversations, emails, through ii) logoed pens, T-shirts etc. worn at conferences or workshops, through iii) storytelling of previous successful experiences and through iv) frequent occurrences that meld the realms of work and leisure.<br>8) Create continuity of collaboration thereby creating a shared culture. Enable virtual teams to focus on reacquiring norms and attitudes over time and renegotiating the meaning of these norms and attitudes when change takes place.<br>9) Provide shared incentives, shared |

| | | |
|---|---|---|
| | | rewards and shared penalties.<br>10) Draw upon past experience to handle new tasks. |
| **Process**<br>**- Mediating tools (collaborative practices)** | ▪ Experimental<br>▪ Flexible/agile and disciplined<br>▪ Improvisational<br>▪ Extensive use of prototypes<br>▪ Bug driven development<br>▪ Maintenance as evolutionary redevelopment<br>▪ Competing technical solutions<br>▪ Power-user communities | 1) Welcome healthy competition and allow parallel solutions to compete.<br>2) Encourage temporary co-location of developers. Frequent face-to-face check points are important.<br>3) Combine flexibility/agility with structure/discipline.<br>4) Ensure high level of planning with minimal structure.<br>5) Use high-level milestones and deliverables to create momentum but be ready to change them and improvise at the spur of the moment.<br>6) Employ a rigorous development process that allows fast release cycles, fast modifications and fast feedback from the users.<br>7) Avoid "Big Bang" integration of components. Do small changes and frequent releases. Concurrent testing is also vital.<br>8) Cultivate "power users communities" to support local users and help developers. Pick such power users wisely.<br>9) Encourage turnarounds in staffs' job posts in order to minimise problems due to loss of expertise.<br>10) Ensure and establish good and extensive communication flows between different layers in the collaboration. |

Table 8.1 Framework for engendering HEAD_SDC: Lessons for future GSD projects

## 8.3 Chapter summary

This chapter combined the analytic arguments of the previous chapter with activity theoretical discussions on collaborative activities and literature on GSD, agile development, collaborative practice (Knorr-Cetina 2001), communities (Wenger 1998; Porra 1999) and organisations (Orlikowski 2000) in order to create a framework based on the most important lessons learned while performing Grid development in the PP context. The framework, primarily emergent from the PP Grid development case study is enhanced by AT and focuses on concepts such as sense-making, improvisation, bricolage and collective mindfulness employed inductively both to hone the framework and in order to reflect its relevance for GSD practice.

Constructing a Grid reflects the challenges of infrastructure development and is contingent and uncertain. The development of a Grid (like any information infrastructure) is rarely built in an entirely top-down, orderly, blueprint way **(**Ciborra and Hanseth 1998). Within this Grid development there are “shared patterns, processes and emergent lessons” (Edwards, Jackson et al. 2007), and it is the primary scope of the framework developed here to incorporate the lessons learned in order to provide concrete practical recommendations for those considering the collaborative development of large-scale distributed systems.

The framework is built around three central components, the structure, history and culture, and process elements, which characterise the design of any sound organisation (Orlikowski 2002). It is hoped that the framework will prove useful for both researchers attempting to better understand collaborative distributed systems development and for those involved in funding, managing and participating in such GSD projects. The framework is reflective and resonant and should not be seen or treated as prescriptive canonical rules; rather it is hoped that it will inform present systems development literature by describing new “forms” or guidelines for distributed systems development practice.

The next and final chapters of this thesis summarise the research and present the general conclusions of the PhD.

# 9 Conclusions and future research

## 9.1 Introduction

This research studied a unique and "exceptional" collaborative context of Grid distributed development. It examined the attempt of particle physicists with extensive knowledge in computing to develop a usable Grid for processing, analysing and sharing experimental data, in their effort to understand the universe. These people acted in unpredictable and original ways and through this, new forms of practices and new forms of innovations were born. This research thus tried to interpret the physicists' practices by particularly focusing on their collaboration, to come up with plausible coherent stories of relevance to the wider context of e-Science, GSD, agile development and similar novel mega-projects.

This chapter begins by presenting a brief synopsis of the entire dissertation and its key arguments (Section 9.2). Section 9.3 follows reviewing the research questions identified in the first chapter briefly discussing how they have been addressed, while Section 9.4 outlines and discusses the theoretical and practical contributions of the work. Section 9.5 discusses the research's limitations followed by Section 9.6 providing some suggestions for future research. Section 9.7 finally presents some concluding remarks of the thesis.

## 9.2 Overview of the thesis

*Chapter 1* frames the study as a case of a large-scale distributed systems development situated within an "exceptional" collaborative context. The main research questions, emerging from the perceived gap in the literature, are identified justifying the relevance of the case as well as the methodological and theoretical choices adopted.

*Chapter 2* reviews literature on GSD, open source development, agile development and collaborative practice with a specific focus on communities – upon which an understanding of the problem can be derived. Such literature reinforces the theoretical model employed as well as provides the conceptual basis upon which the framework of guidance is founded.

*Chapter 3* defines the theoretical lens that facilitated the analysis and explanation of the empirical findings of the research. GSD is understood to be a human collaborative activity so therefore AT is adopted to analyse the complex dynamics, interactions and contradictions manifested within such an activity. This chapter presents a detail commentary on AT by demonstrating its developmental principles and its suitability as the main theoretical framework of this thesis. The four main AT conceptual tools applied in the research are further explained and discussed in depicting their appropriateness to facilitate the analysis of the case.

*Chapter 4* presents the methodological decisions under which the empirical study was approached and operationalised. This involves a brief explanation of the three main philosophical strands of scientific enquiry – positivism, interpretativism and critical research – and subsequently discusses the justification of the choice of interpretative research. This is then followed by a detailed research design that provides the structure and the basis for the operationalisation of the study. The design consists of justifications for a qualitative case-study strategy, the qualitative evidence and how these were gathered through several qualitative data collection techniques, such as semi-structured interviews, informal conversations and documents. Finally, the techniques and tools used in interpreting the data are briefly presented.

*Chapter 5* describes the case under study. The LHC particle accelerator, the Grid's technical elements as well as the three key projects representing the main focus of the research are discussed. This chapter should be seen as complementary to Chapter 6, in providing a historical context of the projects under study.

*Chapter 6* provides the empirical findings of the research. It presents a thorough discussion of how the Grid is collaboratively developed by various distributed actors with different goals, priorities and interests, but still acting as a collective. The chapter consists of two main parts. The first part describes the structure and management of the collaborative development effort pointing to the interesting and distinctive characteristics of the PP community such as the shared goals driving the work, freedom, trust, consensus and democratic decision making. The second part describes the development process with a particular focus on the practices employed and tools used. The findings show that formal structured methodologies do not play a key role in their GSD; improvisation, experimentation, trial and error and emergent change rather than methodological behaviour and planned activities are fundamental aspects of their development process.

*Chapter 7* provides a detailed analysis of the data through the conceptual tools of AT. The distributed development activity is portrayed as an overarching activity system composed of three networked interlinked sub-activity systems that interact, compete and contradict in order to fulfil their objectives. The mutually reinforcing and contradictory relationships between the elements of this overarching activity system and between the elements of the networked interacting sub-activity systems create tensions, conflicts and imbalance. The resolution of such contradictions, although such contradictions cannot always be resolved, is achieved through a series of "implicit" expansive learning cycles and gives rise to new collaborative practices and processes that foster the global development of the Grid. It is in these new practices that valuable insights are gained.

*Chapter 8* provides a thorough discussion of the findings and analysis undertaken in Chapters 6 and 7. Such a discussion reveals a new type of community, which the author defines as a HEAD_SDC, which is created to support, manage, organise and structure the Grid's development. This study suggests that this different type of community, exhibits additional characteristics to the ones defined in a traditional CoP (Wenger 1998); rather this community is highly experimental, goal-oriented and directed, and it is here suggested to be understood as a "collaboration of innovation". This chapter, therefore, takes a step further to propose and develop a framework of guidance that provides practical recommendations to other organisational contexts performing GSD as to how such a community can be engendered. This framework represents the major contribution of this thesis to the advancement of ISD.

## 9.3 Research questions and outcomes

The outcomes of the key research questions identified are here presented and discussed.

*(1) What is the nature of systems development for such a large-scale global project?*

This study shows that systems development in such a global project is more emergent, more continuous, more difficult to control and more affected by people's tradition and culture. Collaborative relationships in such a project are negotiated in "an ongoing communicative process" (Hardy, Phillips et al. 2003), emerge from informal relationships and challenge the traditional emphasis on formal agreements with identified goals, rational partner selection and performance monitoring (Powell, Koput et al. 1996) aligned with traditional plan-based approaches to systems development practice (Avison and Fitzgerald 2003a). Although as a large distributed project the LCG faces issues and uncertainties similar to such projects, nevertheless it is interesting to focus on their response to these challenges, which is not to employ traditional approaches, management structures and methods but rather to improvise and be flexible and agile. They do not, however, employ already established agile methods such as XP, or Scrum. Intriguingly, as particle physicists argue, any kind of method is found to be constraining in such a context, therefore they rather bring improvisation and emergent change to the forefront with minimal organising and therefore exhibit a spirit of agility (Lyytinen and Rose 2006) to enable them to better face the demands of the distributed environment.

This is in part achieved by minimising the clear pre-defined plans or systematic methods guiding the development of the Grid and by emphasising on the skills and competences of the people involved. Such Grid development relies on ad-hoc solutions and has a strong sense of experimentation and trial and error matched with pragmatic problem solving, thereby enabling nimbleness. The Grid's development process therefore emerges out of their adaptation and experimentation with changes, breakdowns, multiple meanings, changing requirements and opportunities (Orlikowski 1996). Such flexible, experimental and improvisational attitude towards development is facilitated by the established structures and processes of the community as well as by an already existing culture, history and tradition that appreciates such experimentations and considers them as legitimate learning experiences, which can lead to innovation. The agility demonstrated in such a large-scale and global project is thus reflective of multiple PP collaborative organisational practices.

This study also suggests that the nature of systems development can be seen as an exercise of co-aligning the technical and social elements of a GSD project. This research indeed demonstrates that the Grid development advances as its PP users actively reconfigure both their work practices and analytical tools in response to the Grid. They are not passive in this alignment process, but rather, as a community they have an equally strong influence in shaping the form of the Grid that they use.

*(2) What practices emerge in response to the demands of such development?*

The practices and processes that emerge have been thoroughly discussed in the previous chapters. Such pragmatic, experimental and agile practices evolve and become better through "implicit" expansive learning cycles taking place throughout the Grid's development. The various contradictions, conflicts, competitive relationships between the people and projects involved in such a development as well as the continual battles between agility and rigor are just some of the reasons why people have to reconstruct their understanding and knowledge and collectively develop new more flexible practices to address the challenges involved in distributed development. This study shows that innovative systems development activities may take place outside formal projects as creative members of the community learn about new opportunities, seek better, quicker and "fancier" algorithms, react to emergent requirements and respond to unexpected problems. These activities are less manifest and more difficult to grasp than in formal development projects with steering committees, project managers and fixed milestones, but are the ones that seem to demonstrate agility and are thus found to be equally important.

*(3) How do these practices influence the development of large-scale and global systems?*

This research demonstrates that the collaborative systems development practices employed or emerged turn the development into a fundamentally sense-making process (Weick 1999) where understanding and knowing lies in the path of action (Orlikowski 2002). This thesis reinforces Weick's (2001) concept of sense-making as it demonstrates that the subjects strive to convert a world of experience into an intelligible and meaningful world by simultaneously discovering and acting, with their actions effecting what it is already discovered. It is by developing prototypes that developers discover what their emergent design means and where they are heading.

The study shows that the strong core of experimentation and trial and error transforms the systems development process into an act of interpretation and attention rather than an act of planning or decision-making (Pant and Hsu 1995). Improvisation, in order to make sense of unexpected possibilities and constraints, is what guides their actions. Developers are never in control of the development process, rather are continually challenged by having to address the emerging requirements and the conflicting priorities. They are thus forced to revise their sense of what is happening and what needs to be accomplished, which promotes attention rather than intention to become central to the development process of such a large-scale distributed system.

*(4) What lessons are learned from the development of the PP Grid that can be translated into the wider field of GSD?*

It is in all of the above that valuable lessons are found that may be translated into the wider context of GSD. These lessons, which have already been discussed in the analysis and discussion chapters, give rise to a new framework of guidance, which is both theoretically founded and practically informed, and which provides guidance and practical recommendations towards engendering HEAD_SDC within GSD projects to facilitate distributed development activities and work. This framework informs agile development literature (Agerfalk and Fitzgerald 2006) by describing how such a global HEAD_SDC exhibits agility and contributes to the debate centred around GSD urgently needing strategies, practices and guidelines for effectively developing and managing large-scale systems in a global way (Smite, Wohlin et al. 2010).

The following section provides the contributions of this study.

## 9.4 Research contributions

The thorough discussion and analysis of the empirical findings allows for the identification of a number of “original” contributions both to theory and practice. “Original” here denotes new knowledge contribution that has an explicit furtherance of existing knowledge. Some of the contributions developed or identified here are substantial, while others are merely validations of other research findings in the particular context of this thesis, mirroring the situation described by Kallinikos (1999) with reference to the contributions of his study of the computerisation of a dairy plant.

### 9.4.1 Contribution to practice

As the IT industry becomes increasingly globalised there is an increasing necessity to develop products and services in large partnerships across diverse communities (Damian and Moitra 2006). Traditionally such relationships (for example outsourcing) have necessitated contractual obligations and hierarchical control (Carmel and Tjia 2005). Yet for some global activities this is inappropriate and it is widely argued that new models of collaborative systems development practices must be discovered (Smite, Wohlin et al. 2010). It is to this end that the contribution of this research provides a framework of guidance outlining the lessons from the PP community's systems development practices.

The major contribution of this research is the development of a framework of guidance (extensively discussed in Chapter 8) for fostering the engendering of, what the researcher defines as HEAD_SDC, which can support and facilitate GSD. The framework, primarily emerging from the PP Grid development case study, is also influenced by AT and focuses on concepts of sense-making, improvisation and collective mindfulness. The framework is based on three important dimensions – the structure, history/culture and process – which have been identified from the literature (Robey 1991; Nohria 1995) and from AT (Engestrom 1987) and provide the basis for the design of any sound organisation (Orlikowski 2002).

The framework of guidance introduces a hybrid community that exhibits additional characteristics than the ones already defined in a traditional CoP or a colony (Wenger 1998; Porra 1999). Rather, this community is experimental, agile, clustered and virtual, with limited lines of authority and highly directed and motivated by shared goals and sacred causes. This research shows that the Grid's development in this community is visionary, experiential, passionate, agile and emergent.

This framework enables others to elaborate and explore elements often pushed into the background in the discussions of GSD, such as environmental conditions, individual skills, professional cultures, organisational structures, communication patterns and interpersonal relationships. Through the framework, a set of practical recommendations is proposed, reflecting those organisational practices from the case, and highlighting the means by which such a community may be engendered to facilitate GSD.

The framework offers instructive lessons to other domains attempting to construct large-scale systems in a distributed fashion and it is hoped that it will prove useful for both researchers attempting to better understand collaborative distributed development and for those involved in funding, managing and participating in such global development projects. This framework is reflective and should be seen as an attempt to fill in the perceived “gap” present in the global systems development literature (Agerfalk, Fitzgerald et al. 2009) urgently needing new system development paradigms, practices and strategies to support the development of distributed systems as well as facilitate the management of, and collaboration within, global projects. It also contributes to agile development literature by describing the conditions under which such a HEAD_SDC exhibits agility on a global scale.

While the HEAD_SDC framework provides the practical contribution of the thesis it also represents this study’s theoretical contribution to theories of collaboration and communities (Wenger 1998; Porra 1999; Weick and Roberts 1993). HEAD_SDC allow for a renewed understanding of collaborative work within a “unique” form of a globally distributed community and thus enriches such literature by identifying distinctive characteristics of such a global virtual community. The underlying conditions of how to engender innovative virtual communities, which can overcome the “capricious” nature (Wenger 1998) of CoP but can still exhibit characteristics of shared identity, common interests and ones that overcome the issues related to trust, and the sharing of knowledge are needed (Kimble and Hildreth 2005). HEAD_SDC, therefore, informs such a literature and contributes to this “gap”.

### 9.4.2 Contribution to theory

The theoretical framework of this thesis is based on AT (Engestrom, Miettinen et al. 1999), a theory chosen because it explains and foresees the complex dynamics of collaborative activities, by taking into account the environment, the structure, the history and culture, the artefacts, the processes, the motivations and the complexities involved in real-life actions. It therefore allows for the identification of contradictions and focuses on the resulting learning outcomes that emerge from the contradictions' resolution (Crawford and Hasan 2006).

AT has been applied to the design of software systems, and research to date has indicated its usefulness e.g. (Kuutti 1996; Korpela, Soriyan et al. 2000; Crawford and Hasan 2006; Mursu, Luukkomen et al. 2007). However, to the author's knowledge, this research represents one of the first attempts to apply AT for studying large-scale distributed systems development and understanding the global collaborative efforts of widely distributed developers. AT is still widely ignored outside Scandinavian IS circles and therefore has not yet become a popular theory in mainstream IS research (Engestrom 2001). This research demonstrates the value of employing AT in researching GSD showing its usefulness for exploring the inherent contradictions of such a process and should be seen as an attempt to inform the wider IS research regarding the theory's advantages.

Nevertheless, the theory is not without limitations (Roth 2007). Various authors argue for the need to better account for identity, power relations (Daniels and Warmington 2007) and multi-motivation in activities (Kaptelinin and Nardi 2006). Engestrom (2001) has proposed a third generation of AT in an attempt to respond to existing criticism to enhance the "dialogue, multiple perspectives and voices and network for interacting activity systems (ibid). This third generation of AT, nevertheless, remains under-theorised, with a limited number of papers demonstrating its effectiveness, particularly in the field of GSD. Engestrom (2009) calls for new conceptual tools to further enhance the theory and ones which can help make sense of this globally distributed and "online" Web 2.0 generation we live in.

This thesis provides such much-needed evidence, demonstrating the value of complex networked AT, and demonstrating its importance in analysing such distributed and virtual collaborative systems development activities. Crucially, however, the thesis' adoption of the activity triangle (second generation AT) in structuring the analysis of such a messy development situation led to a new recursive structure for activities within AT. The thesis identifies an overarching "networked activity system" – defined by the traditional activity triangle, but, and in contrast to traditional AT, considers its achievement to consist of networked interacting sub-activity systems sharing the same object (hence the possibility of an overarching networked activity system). The sub-activity systems whilst having differing motivations, tools, rules, division of labour and community, yet, the overarching activity system's components present a merge of the tools, rules, division of labour and community defined by the networked sub-activity systems (or the strongest sub-activity system imposes its own rules, tools, division of labour and community). Tensions between the overarching activity system's components therefore exist (imposed, institutionalised, etc.), emerging from conflicts between the networked sub-activity systems that interact, differ, compete and contradict in order to fulfil their own priorities and motivations.

Based on this empirical study it is suggested that the third generation of AT could be enhanced to address the complexity of such distributed cases. This thesis concludes that the direction taken by the proposed third generation of AT might be inappropriate for collaborative distributed systems development studies such as this one. The increased complexity of such distributed activities, suggests that future developments of AT should be based on this recursive development of the second – in which networked accounts are undertaken. It should be based on the already existing activity second generation framework but with the following recommended additions, which are the identified networked sub-activity systems which interact, compete and contradict and might visibly or invisibly work towards the fulfilment of the shared object. This thesis therefore proposes a substantive extension to AT that furthers our understanding of collaborative global systems development activities by developing what the author defines as an "AT meta-framework 2.5" reflecting on ideas of the third generation but also contributing to the third generation by enhancing the second generation of AT.

## 9.5 Limitations of the research

The previous section discussed the contributions of this research to theory and practice. However, this study also comes with a number of limitations that are important to acknowledge.

The first limitation is derived from the distributed nature of this study, which at times made this research challenging. It is indeed difficult to study such a widely distributed and global community such as the PP where the researcher has to constantly travel across Europe in order to meet with separate contributors, attend meetings and perform observations. As a result, there were times, because of funding and resource limitations, that it has been impossible to meet with certain people who were involved in systems development and held a strong understanding of the work being done. Furthermore, the distributed nature of the study made it necessary to focus on the UK and CERN in particular and therefore much of the research is founded on the LCG, GridPP and EGEE organisational projects and contexts, describing their practice, and inevitably leaving other important projects, such as the Italian project's contribution to LCG, or the American project's contribution to LCG etc. aside. However, this limitation was arguably resolved by the multiplicity of roles that individuals held in different projects. This means that the researcher had interviewed people that at some point were involved in and/or were funded by projects other than GridPP, EGEE or LCG and therefore could provide a thorough interpretation on how the global collaborative development work was performed based on their previous knowledge and understanding.

The development of the Grid is performed by a diverse group of people, coming from different disciplines and backgrounds, including both computer scientists and particle physicists. As evident by the findings, the PP tradition, culture and institutional practice prevailed. PP practices are reflected both in the way the Grid is being developed and in the way the collaborative effort is organised. Although this research explored the PP practices, being a study of collaborative distributed systems development meant that all accounts and interpretations should be taken into perspective and presented in the thesis, which at times presented a challenge, since it was difficult to find a balance when describing what was going on in practice. One of the limitations of this study is the dominance of the PP culture that made it difficult to critically evaluate the "voice" of computer scientists involved in the project. At times this thesis may appear overly geared towards the practices of particle physicists. This is partly a methodological problem because those computer scientists involved are so indoctrinated to the practices of the PP community, something that makes the vocal dominant view of the PP community become more obvious.

Researching such a bright, reflexive and visible community indeed presented a challenge. Their intelligence means that they interpret what they say and what they do themselves and try to come up with their own frameworks and ideas of the way they work based on what they believe the researcher would like to hear. Their desire to be seen as successful and innovative means that their interpretation of what they do is also entwined with their desire to make themselves sound good. This therefore made it hard to cut through their interpretations.

This study is not observing laboratory rats, rather intelligent scientists attempting to explore the universe, which at times was challenging as the researcher was also drowned into this grand vision of particle physicists. Travelling to CERN was like travelling in a 'Mayblin' world of streets named after Oppenheimer and Einstein; seeing an experiment that was ten stories high was like something coming out of a science fiction story and as an interpretative researcher, the author, understands and acknowledges that at times she was affected by that. However, the researcher was still able to remain distanced and be critical towards what the interviewees argued and what she observed.

A further limitation of this study, one might argue, is the extent to which the lessons drawn from the community's systems development practices under study are relevant and generalisable to other contexts. Clearly, the unique and obscure nature of the community under study is a limitation to the study. The aim of this research, as an interpretative qualitative research in the IS context, however, is to come up with plausible coherent stories of what is happening and therefore to understand the underlying structure of the phenomena under study and use this knowledge to 'inform' other settings, rather than normatively interpret the world (Checkland and Scholes 1990). This research explores an unusual systems development context and in light of this, some interesting lessons have been drawn and provided as practical recommendations within the framework of guidance developed here. The practical recommendations provided here should not be seen or treated as prescriptive canonical rules but hopefully they will allow others involved in such distributed virtual development projects to reflect on their practice and their context in light of them. While it is certainly appreciated that this case does not present an ideal form of distributed systems development, it still adds to the studies of GSD and agile development by providing a somewhat different perspective to development work with organisational practices stemming from a non-bureaucratic organisation.

## 9.6 Future research

This final section of the thesis reflects on how the work developed and presented here can guide future research. Some of the limitations discussed above present further challenges for future research endeavours into large-scale distributed development of distributed systems.

This research has been taken forward into the following directions. The researcher has helped secure funding with Oxford University to study the information use practices of scientists in the physical sciences. A particular focus of this research is the PP community and their interaction with the Grid. It was impossible, given the time, scale and scope limitations of a PhD thesis, to explore effectively how users interacted with the Grid. This is, therefore, something that at the moment is being pursued.

Similarly, it would be interesting to extend this study and the study's outcomes to other types of organisations and communities that deal with distributed development. A cross-sector study, examining the systems development practices in other global organisational contexts or communities – such as open source development projects, global sourcing projects or the medical or biology communities – could investigate the issues related to attitudes, norms and practices that impact or facilitate similar or different behaviours in approaching global development.

Another interesting and important aspect that this PhD research could not address as it would make the research's scope broad is the focus on the collaborative working practices involved when such a large-scale Grid infrastructure moves from being developed to become production. Therefore, exploring the working practices of various global agencies and actors during the Grid's deployment and maintenance. Such explorations open further research avenues that could contribute to the debates within IS which has largely ignored the concept of IS deployment as well as to the debates on IS implementation (Walsham and Kwong Han 1993; Avgerou and Cornford 1998) extending this area's generally traditional view of systems implementation into the concerns of a globally distributed information infrastructure development and deployment. Relevant and important lessons can be drawn from such a study and can be provided to a range of different industries engaged in this work.

The application of Grid technology is quite recent. Therefore, it is exciting to carry out research into this area. While the working practices involved in deploying and maintaining such large-scale infrastructures is ripe for future research, a further research avenue would be to study the end-users' reaction towards the final system as the Grid becomes a commercial end-product. Using an alternative theoretical framework such as attempting a social construction of technology analysis (Pinch and Bijker 1984; Orlikowski and Gash 1994) can be useful, exploring the congruence and/or incongruence in the technological frames of users regarding the final output and how a closure and stabilisation can be achieved in order for the Grid to become an institutionalised tool embedded in their everyday practices.

Exploring the joint study of "managerial" intervention and users' appropriations in the post-implementation period of the Grid and of other complex systems, therefore, provides an exciting opportunity, which is argued to remain an under-explored area (Robey, Im et al. 2008). While a great deal of research literature primarily focuses on phenomena related to adoption (Jasperson, Carter et al. 2005) models that support management for controlling and shaping the implementation (Iacovou, Benbasat et al. 1995; Gallivan and Depledge 2003; Teo, Wei et al. 2003), it is argued that it largely neglects the agency of managers and users during the post-implementation period and therefore make this topic interesting to examine. This topic is particularly relevant in the context of inter-organisational IS, where the practitioner literature shows that initial adoption does not necessarily lead to assimilation (Jap and Mohr 2002; Teo, Ranganathan et al. 2006).

## 9.7 Concluding remarks

This research has been an exciting and sometimes challenging intellectual journey to investigate the collaborative development of large-scale systems, of which the Grid was taken as an example, in a global way.

This thesis is the outcome of extensive and in-depth investigations of the global collaborative systems development practices employed and emerged during the Grid's development within a real-life widely distributed community. This study is about "distinctive" collaborative distributed infrastructure systems development. The motivation throughout the study was to explore this phenomenon within the context of the PP community, a reflective and bright community with a "distinctive" tradition and history in computing, in order to unveil lessons for more efficient and flexible systems development practices and processes.

Although the limitations and future research above may leave some problems unaddressed, the main objectives of this research as stated in Chapter 1 and evolved throughout the thesis, have been thoroughly addressed, discussed and conceptualised and have been grounded on objective interpretations and scientific analysis of the findings. The goal of a PhD research is not to study the whole world, but that tiny portion of the world studied here has achieved an important contribution to progress in the area of collaborative GSD.

# 10 References


Abbot, M. (2009). A new path for science? In: The fourth paradigm: Data-intensive scientific discovery. T. Hey, S. Tansley and K. Tolle (Eds), Redmond, Washington, Microsoft Research.

Abrahamsson, P., J. Warsta, et al. (2003). New direction on agile methods: A comparative analysis. 25th International Conference on Software Engineering (ICSE'03).

Adams, M., D. Edmond, et al. (2003). The application of activity theory to dynamic workflow adaptation issues. 7th Pacific Asia Conference on Information Systems Adelaide, South Australia.

Adolph, S. (2006). What lessons can the agile community learn from a maverick fighter pilot. AGILE 2006 Conference, Minneapolis, Minnesota, USA, IEEE Computer Society

Agerfalk, J. P. and B. Fitzgerald (2006). "Flexible and distributed software processes: Old petunias in new bowls?" Communications of the ACM **49**(10).

Agerfalk, J. P., B. Fitzgerald, et al. (2009). "Flexible and distributed information systems development: State of the art and research challenges." Information Systems Research **20**(3): 317-328.

Ahuja, M. K. and J. E. Galvin (2003). "Socialization in virtual groups." Journal of Management **29**(2): 161-185.

Amendolia, S. R., F. Estrella, et al. (2004). Grid databases for shared image analysis in the mammoGrid project. 8th International Database Engineering & Applications Symposium, Portugal.

Anderson, M. (2008). "How the large hadron collider might change the web: The LHC computing Grid may teach the internet how to quietly handle reams of information." Scientific American.

Angell, I. O. and S. Smithson (1991). Information systems management: Opportunities and risks. London McMillan Press.

Armbrust, M., R. Griffith, et al. (2010). "A view of cloud computing." Communications of the ACM **53-58**(4)

Armstrong, D. J. and P. Cole (1995). Managing distances and differences in geographically distributed work groups. In: Diversity in work teams: Research paradigms for a changing workplace. S. E. Jackson and M. N. Ruderman. Washington DC, American Psychological Association.

Avgerou, C. (2000). "Information systems: What sort of science is it?" Omega, International Journal of Management Science **28**: 567-579.

Avgerou, C. (2002). Information systems and global diversity. Oxford, Oxford University Press.

Avgerou, C. (2005). "Doing critical research in information systems: Some further thoughts." Information Systems Journal **15**: 103-109.

Avgerou, C. and T. Cornford (1998). Developing information systems: Concepts, issues and practice. Basingstoke, UK, Macmillan.

Avison, D. and B. Fitzgerald (2003a). Information systems development: methodologies, techniques and tools. UK, McGraw Hill.

Avison, D. and G. Fitzgerald (2003b). "Where now for development methodologies?" Communications of the ACM **46**(1): 79-82.

Bajec, M., M. Krisper, et al. (2004). The scenario for constructing flexible, people-focused systems development methodologies. European Conference of Information Systems.

Baker, M., R. Buyya, et al. (2002). Grids and Grid technologies for wide-area distributed computing. In: Software Practice and Experience, John Wiley & Sons.

Bandyopadhyay, S., S.R. Marston, et al. (2009). "Cloud computing: The business perspective." SSRN eLibrary.

Bansler, J. P. and E. Havn (2004). Improvisation in information systems development. IFIP WG 8.2 Conference, Manchester, England.

Barab, S. A., M. Barnett, et al. (2002). "Using activity theory to understand the systemic tensions characterizing a technology-rich introductory astronomy course." Mind, Culture, and Activity: An International Journal **9**(2): 76-107.

Bardram, E. J. (1997). Plans as situated action: An activity theory approach to workflow systems. 5th European Conference of Computer Supported Coopeartive Work, Lancaster, UK, Kluwer Academic publishers.

Bardram, E. J. (1998). Designing for the dynamics of cooperative work activities. Computer Supported Cooperative Work Conference, New York, ACM Press.

Barnes, B. (2000). Understanding agency: Social theory and responsible action. London, Sage publication.

Barnes, B. (2001). Practices as collective actions. In: The practice turn in contemporary theory. T. R. Schatzki, K. Knorr-Cetina and E. Von Savigny.

Barthelmess, P. and K. M. Anderson (2002). "A view of software development environments based on activity theory." Computer Supported Cooperative Work **11**(1-2): 13-37.

Baskerville, R. L. and J. Pries-Heje (2004). "Short cycle time systems development." Information Systems Journal **14**: 237-264.

Battin, R. D., R. Crocker, et al. (2001). "Leveraging resources in global software development." IEEE Software **70**(7).

Becker, J. and B. Niehaves (2007). "Epistemological perspectives on IS research: A framework for analysing and systematising epistemological assumptions." Information Systems Journal **17**(2): 197-214.

Bell, J. (1993). Doing your research project: A guide for first-time researchers in education and social science. Buckingham, Open University Press.

Benbasat, I., D. K. Goldstein, et al. (1987). "The case research srategy in studies of information systems." MIS Quarterly **11**(3): 396-386.

Berger, P. L. and T. Luckmann (1966). The social construction of reality: A treatise in the sociology of knowledge. New York, Penguin books.

Berman, F., F. Geoffrey, et al. (2003). The Grid: past, present, future. In: Grid computing: Making the global infrastructure a reality. F. Berman, F. Geoffrey and T. Hey, John Wiley & Sons.

Berners Lee, T. and M. Fischetti (1997). Weaving the web. San Fransisco, CA, Harper.

Berntsen, K. E., J. Sampson, et al. (2004). "Interpretive research methods in computer science." http://se.informatic.uni-oldenburg.de/lehre/sose2005/SS04-Seminar-Methoden.pdf, Retrieved: 20th March 2008.

Bertelsen, O. W. (2000). "Design artifacts: Towards a design-oriented epistemology." Scandinavian Journal of Information Systems **12**: 15-28.

Bertelsen, O. W. (2002). Reorienting. In: Social thinking, software practice. Y. Dittrich, C. Floyd and R. Klischewski.

Bertelsen, O. W. (2003). Contradictions as a tool in IT-design: Some notes. 8th European Conference of Computer Supported Cooperative Work (ECSCW), Helsinkin, Finland.

Bertelsen, O. W. and S. Bodker (2000). "Information technology in human activity." Scandinavian Journal of Information Systems **12**: 3-14.

Bodker, S. (1989). "A human activity approach to user interfaces." Human Computer Interaction **4**(3): 171-195.

Bodker, S. (1991). Activity theory as a challenge to systems design. In: Information systems research: Contemporary approaches and emergent traditions. H. E. Nissen, H. Klein and R. Hirschheim. North Holland.

Bodker, S. (1996). Applying activity theory to video analysis: How to make sense of video data in human-computer interaction. Context and consciousness: Activity theory and human-computer interaction. B. Nardi. Cambridge, Mass: MIT Press.

Boehm, B. and R. Turner (2004). Balancing agility and discipline: A guide for the Perplexed. Boston, Addison-Wesley.

Bohman, J. (1999). Practical reason and cultural constraint: Agency in Bourdieu's theory of practice. In: Bourdieu: A critical reader. R. Shusterman. Oxford, Blackwell**:** 128-152.

Boland, R. J. and R. A. Hirschheim (1987). Critical issues in information systems research. Chichester, Great Britain, John Wiley & Sons.

Bonoma, T. V. (1985). A case study in case research: Marketing implementation. Working paper #9. Boston, Harvard University Graduate School of Business Administration.

Bourguin, G., A. Derycke, et al. (2001). "Beyond the interfaces, Co-evolution inside interactive systems: A proposal founded on the acivity theory." People and Computers: 297-310.

Bourhis, A. and L. Dube (2010). "Structuring spontaneity: Investigating the impact of management practices on the success of virtual communities of practice." Journal of Information Science **36**(2): 175-193.

Bratteteig, T. and J. Gregory (2001). Understanding design. IRIS conference, University of Bergen.

Brede-Moe, N. and D. Šmite (2008). "Understanding a lack of trust in global software teams: A multiple-case study." Software Process: Improvement and Practice **13**(3).

Bremer, J., M. Campolargo, et al. (2002). "Complex Problem solving including Grid and research netwrorking infrastructures." ftp://ftp.cordis.lu/pub/ist/docs/grid/irg-grid_rni-140502.pdf Retrieved 22 February 2007.

Breu, K. and C. Hemingway (2004). "Making organisations virtual: The hidden cost of distributed teams." Journal of Information Technology **19**(3): 191-202.

Britton, D. (2003). GridPP: A project overview. UK e-Science All hands Meeting conference, On behalf of the whole GridPP collaboration.

Britton, D., A. J. Cass, et al. (2009). "GridPP: the UK Grid for particle physics " Philosophical Transactions of the Royal Society A **367**: 2447-2457.

Britton, D. and S. Lloyd (2004). A Grid for particle physics: Managing the unmanageable. All hands meeting Conference, On behalf of the whole GridPP collaboration.

Britton, D. I., S. Burke, et al. (2006a). GridPP: From prototype to production. GridPP, On behalf of the whole GridPP collaboration.

Britton, D. I., S. Burke, et al. (2006b). Performance of the UK Grid for particle physics. GridPP, On behalf of the whole GridPP collaboration.

Broadbent, M. and P. Weill (1999). "The implication of information technology infrastructure for business-process-redesign." MIS Quarterly **23**(2): 159-182.

Brooks, F. (1987). "No silver bullet: Essence and accidents of software engineering." IEEE Computer Magazine: 10-19.

Brown, J. S. and S. Cook (1999). "Bridging epistemologies: The generative dance between knowledge and organizational knowing." Organization Science **10**(4): 381-400.

Brown, J. S. and P. Duguid (2001). "Knowledge and organization: A social practice perspective." Organization Science **12**(2): 198-213.

Bryman, A. (2004). Social research methods, Oxford university press.

Burn, J., P. Marschall, et al. (2002). E-business strategies for virtual organizations. Oxford, Butterworth-Heinemann.

Butler, T. (1998). "Towards a hermeneutic method for interpretive research in information systems." Journal of Information Technology **13**: 285-300.

Butler, T. and B. Fitzgerald (1997). An empirical model of the information systems development process. 5th International Conference on Information System Methodologies: Training and Education of Methodology Practitioners and Researchers, Springer-Verlag.

Carmel, E. (1999). Global software teams: Collaborating across borders and time zones. USA, Upper Saddle River, NJ : Prentice Hall.

Carmel, E. and R. Agarwal (2001). "Tactical approaches for alleviating distance in global software development." IEEE Software **18**(2): 22-29.

Carmel, E. and P. Tjia (2005). Offshoring information technology sourcing and outsourcing to a global workforce. Cambridge, Cambridge University Press.

Carr, N. (2005). "The end of corporate computing." MIT Sloan Management Review **46**(3): 67-73.

Carr, N. (2008). The big switch: Rewiring the world, from edison to google, W. W. Norton & Co.

CERN-newsletter (2007). The LCG project. CERN, IOP Publishing. **42**.

CERN (2004). Infinitely CERN: Memories from fifty years of research, Suzzane Hurter Editions.

CERN. (2010). http://public.web.cern.ch/public/en/About/About-en.html Retrieved 18 April 2008.

Chatterjee, S. (2002). Converger networks and the Grid: 21st infrastructure challenges, American Conference on Information Systems (AMCIS'02).

Che, H. and D. Zhao. (2005). "Managing trust in collaborative software development." http://l3d.cs.colorado.edu/~yunwen/KCSD2005/papers/che-trust.pdf Retrieved 2 May, 2007.

Checkland, P. and J. Scholes (1990). Soft systems methodology in action. Chichester, John Wiley & sons.

Cheng, L., C. R. DeSouza, et al. (2004). "Building collaboration into IDEs." ACM Queue **1**(9): 40-50.

Chetty, M. and R. Buyya (2002). "Weaving computational Grids: How analogous are they with electrical grids?" Computing in Science & Engineering **4**(4): 61-71.

Child, J. (2001). "Trust: The fundamental bond in global collaboration." Organizational Dynamics **29**(4): 274-288.

Chompalov, I., J. Genuth, et al. (2002). "The organisation of scientific collaborations." Research Policy **31**: 749-767.

Chudoba, M. K., E. Wynn, et al. (2005). "How virtual are we? Measuring virtuality in a global organisation." Information Systems Journal **15**: 279-306.
Ciborra, C. (2002). The Labyrinths of information: Challenging the wisdom of systems. Oxford, UK, Oxford University Press.
Ciborra, C. (2004). Digital technologies and the duality of risks, ESRC centre for analysis of risk and regulation, London School of Economics and Political Science.
Ciborra, C. and Associates (2000). From control to drift: The dynamics of corporate information infrastructures. Oxford, UK, Oxford University Press.
Ciborra, C. and O. Hanseth (1998). "From tool to gestell agendas for managing the information infrastructure." Information Technology & People **11**(4).
Cockburn, A. and J. Highsmith (2001). "Agile software development: The people factor." IEEE Computer
Cohen, S. G. and D. E. Bailey (1997). "What makes teams work: Group effectiveness research from the shop floor to the executire suite." Journal of Management **23**: 139-290.
Cole, M. (1988). "Cross-cultural research in the sociohistorical tradition." Human Development **31**: 137-151.
Cole, M. and Y. Engestrom (1993). A cultural-historical approach to distributed cognition. In: Distributed cognitions: Psychological and educational considerations. G. Salomon. New York, Cambridge University Press.
Coles, J. (2005). Grid deployment and operations: EGEE, LCG and GridPP. UK e-Science All Hands Meeting Conference, Nottingham.
Colling, D. J. (2002). GridPP - Developing an Operational Grid. UK e-Science All Hands Meeting Conference, Sheffield.
Collins, G. P. (2008). "The discovery machine." Scientific American **298**(2): 31-37.
Collins, H. M. (2001). What is tacit knowledge? In: The practice turn in contemporary theory. T. R. Schatzki, K. Knorr-Cetina and E. Von Savigny.
Conboy, K. (2009). "Agility from first principles: Reconstructing the concept of agility in information systems development." Information Systems Research **20**(3): 329-354.
Conboy, K. and B. Fitzgerald (2004). Toward a conceptual framework of agile methods: A study of agility in different disciplines. ACM workshop on Interdisciplinary Software Engineering Research (WISER'04), Newport Beach, California.
Cordella, A. (2006). Information infrastructure in action, London School of Economics and Political Sciences, Department of Information Systems.
Cothrel, J. and R. L. Williams (1999). "On-Line communities: Helping them form and grow." Journal of Knowledge Management **3**(1): 54-60.
Crawford, K. and H. Hasan (2006). "Demonstrations of the activity theory framework for research in information systems." Australasian Journal of Information Systems **13**(2): 49-68.
Creswell, J. W. (1998). Qualitative inquiry and research design: Choosing among five traditions, Thousand Oaks, CA: Sage.
Crotty, M. (2003). The foundations of social research. London, UK, Sage Publications.
Crowston, K., J. Howison, et al. (2005). Face-to-face interactions in self-organizing distributed teams. Academy of Management conference, Honolulu, Hawaii.

D'Eredita, M., N. Misiolek, et al. (2005). States of mind as stages of team development: Making sense of strategies for building a virtual team. 5th Annual Hawaii International Conference on Business, Honolulu.

D'Eredita, M. A. and C. Chau (2005). "Coaching from afar: How ubiquitous information technology was used to develop a succesful semi-virtual team." Electronic Journal of Knowledge Management **3**(2): 65-74.

Damian, D. (2003). "Global software development: Growing opportunities, ongoing challenges." Software Process Improvement and Practice **8**: 179–182.

Damian, D. and D. Moitra (2006). "Global software development: How far have we come?" IEEE Software.

Daniels, H. (2005). An introduction to Vygotsky. East Sussex, Routledge.

Daniels, H. and P. Warmington (2007). "Analysing third generation activity systems: Labour-power, subject position and personal transformation." Journal of Workplace Learning **19**(6): 377-391.

Davidow, W. H. and M. S. Malone (1992). The virtual corporation: Structuring and revitalizing the corporation of the 21st century. New York, Harper-Collins.

De Freitas, S. M. and E. Byrne (2006). Activity theory as an analytical tool: A case study of IS development for an anti-retroviral treatment clinic in South Africa. SAICSIT conference.

De Roure, D., N. Jennings, et al. (2003). The semantic Grid: A future e-Science infrastructure. In: Grid computing – Making the global infrastructure a reality. F. Berman, F. Geoffrey and T. Hey, John Wiley & Sons.

De Souza, B. C. (2003). Interpreting activity theory as a software engineering methodology. 8th European Conference of Computer Supported Cooperative Work, Helsinki, Finland.

De Souza, B. C. and F. D. Redmiles (2003). Opportunities for extending activity theory for studying collaborative software development. 8th European Conference of Computer Supported Cooperative Work, Helsinki, Finland.

DeSouza, K. (2003). "Fascilitating tacit knowledge exchange." Communications of the ACM **46**(6): 85-88.

DeSouza, K. and J. R. Evaristo (2004). "Managing knowledge in distributed projects." Communications of the ACM **47**(4): 87-91.

Ditsa, G. (2003). "Activity theory as a theoretical foundation for information systems research." Information Management: Support Systems & Multimedia Technology: 192 - 231.

Doyle, T. (2005). Meeting the particle physics computing challenge, Public Service Review: Department of Trade and Industry.

Dreyfus, L. H. (2001). On the internet: Thinking in action. London and New York, Routledge Press Paperback.

Dutton, J. E., J. M. Dukerich, et al. (1994). "Organizational images and member identification." Administrative Science Quarterly **39**: 239-263.

Edwards, H. K. and V. Sridhar (2002). Analysis of the effectiveness of global virtual teams in software engineering projects. 36th Hawaii International Conference on Systems Sciences (HICSS), IEEE.

Edwards, P., S. Jackson, et al. (2007). Understanding infrastructure: Dynamics, tensions and design.Workshop on History & Theory of Infrastructure: Lessons for New Scientific Cyberinfrastructures, Ann Arbor, University of Michigan

EGEE. (2010). http://project.eu-egee.org/index.php?id=104 Retrieved Retrieved: 20 February 2010.

Engestrom, Y. (1987). Learning by expanding: An activity-theoretical approach to developmental reasearch. Helsinki, Orienta-Konsultit.

Engestrom, Y. (1990). Learning,working and imagining: Twelve studies in activity theory. Helsinki, Orienta Konsultit.

Engestrom, Y. (1997). "Coordination, cooperation and communication in the courts." Mind, Culture, and Activity: 369 - 388.

Engestrom, Y. (1999a). "Expansive visibilization of work: An activity theoretical perspective." Computer Supported Cooperative Work **8**: 63-93.

Engestrom, Y. (1999b). Activity theory and individual and social transformation. In: Perspectives on activity theory. Y. Engestrom, R. Miettinen and P. Raija-Leena. Cambridge, Cambridge University Press.

Engestrom, Y. (1999c). Innovative learning in work teams: Analysing cycles of knowledge creation in practice. In: Perspectives on activity theory. Y. Engestrom, R. Miettinen and P. Raija-Leena. Cambridge, Cambridge University Press.

Engestrom, Y. (2000a). "Activity theory and the social construction of knowledge: A story of four umpires." Organization **7**(2): 301 - 310.

Engestrom, Y. (2000b). "Activity theory as a framework for analysing and redesigning work." Ergonomics **43**(7): 960-974.

Engestrom, Y. (2001). "Expansive learning at work: Toward an activity theoretical reconceptualisation." Journal of Education and Work **14**(1): 133-156.

Engestrom, Y. (2003). The horizontal dimension of expansive learning: Weaving a texture of cognitive trails in the terrain of healthcare in Helsinki. International Symbosium 'New Challenges to Research on Learning', University of Helsinki, Finland.

Engestrom, Y. (2004). "New forms of learning in co-configuration work." Journal of Workplace Learning **16**(1-2): 11-21.

Engestrom, Y. (2007). From communities of practice to mycorrhizae. In: Communities of practice: Critical perspectives. J. Hughes, N. Jewson and L. Unwin. Abingdon, Routledge.

Engestrom, Y. (2009). The future of activity theory: A rough draft. In: Learning and expanding with activity theory. A. Sannino, H. Daniels and K. Gutierrez. Cambridge, Cambridge university press.

Engestrom, Y., R. Engestrom, et al. (2002). Can a school community learn to master its own future? An activity-theoretical study of expansive learning among middle school teachers. In: Learning for life in the 21st century. G. Wells and G. Claxton, Malden, MA: Blackwell.

Engestrom, Y., R. Miettinen, et al. (1999). Perspectives on activity theory. Cambridge, Cambridge university press.

Erenkrantz, J. R. and R. N. Taylor (2003). Supporting distributed and descentralized projects: Drawing lessons from the open source community. California, Institute for Software Research, University of California.

Erwin, L. and B. Jones (2008). "Enabling Grids for e-Science: The EGEE project."

Etzioni, A. and O. Etzioni (1999). "Face-to-face and computer-mediated communities: A comparative analysis." The Information Society **15**: 241-248.

EuropeanCommission. (2006). "Future for european Grids: Grids and service-oriented knowledge utilities." ftp://ftp.cordis.europa.eu/pub/ist/docs/grids/ngg3_eg_final.pdf Retrieved 22/02/2007.

Fang, Y. H. and C. M. Chiu (2010). "In justice we trust: Exploring knowledge-sharing continuance intentions in virtual communities of practice." Computers in Human Behavior **26**(2): 235-246.

Feldman, M. S. and B. T. Pentland (2003). "Reconceptualizing organizational routines as a source of flexibility and change." Administrative Science Quarterly **48**: 94-118.

Feldman, M. S. and A. Rafaeli (2002). "Organizational routines as sources of connections and understandings." Journal of Management Studies **39**(3): 309-331.

Feller, J. and B. Fitzgerald (2000). A framework analysis of the open source software development paradigm. International Conference of Information Systems (ICIS), Brisbane, Australia.

Fitzgerald, B. (1994). Whither systems development: Time to move the lamppost. Second Conference on Information Systems Methodologies. Swindon, BCS Publications.

Fitzgerald, B. (2000). "Systems development methodologies: The problem of tenses." Information Technology & People **13**(3): 13-22.

Fitzgerald, B., N. Russo, et al. (2002). Information systems development: Methods in action. London, McGraw Hill.

Flor, N. V. (2006). "Globally distributed software development and pair programming." Communications of the ACM **49**(10).

Floyd, C. (2002). Developing and embedding autooperatinal form. In: Social thinking, Software practice. Y. Dittrich, C. Floyd and R. Klischewski.

Foot, K. A. (2001). "Cultural-historical activity theory as practice theory: Illuminating the development of a conflict-monitoring network." Communication Theory **11**(1): 56-83.

Foster, I. (2003). The Grid: A new infrastructure for 21st century science. In: Grid computing – Making the global infrastructure a reality. F. Berman, F. Geoffrey and T. Hey, John Wiley & Sons.

Foster, I. and C. Kesselman (2003). The Grid in a nutshell. In: Grid resource management – State of the art and future trends. J. Nabrzyski, J. Schopf and J. Weglarz, Kluwer Academic Publishers.

Foster, I. and C. Kesselman (2004). The Grid 2: Blueprint for a new computing infrastructure, Elsevier.

Foster, I., C. Kesselman, et al. (2001). "The anatomy of the Grid: Enabling scalable virtual organizations." International Journal of Supercomputer Applications **15**(3): 200-222.

Foster, I., C. Kesselman, et al. (2003). The anatomy of the Grid. In: Grid computing – Making the global infrastructure a reality. F. Berman, F. Geoffrey and T. Hey, John Wiley & Sons.

Fowler, M. (2006). Using an agile software process with offshore development, ThoughtWorks consultancy.

Fowler, M. and J. Highsmith (2001). "The agile manifesto." Software Development(August).

Fulk, J. and G. DeSanctis (1995). "Electronic communication and changing organizational forms." Organization Science **6**(4): 337-349.

Furst, S. A., M. Reeves, et al. (2004). "Managing the life cycle of virtual teams." Academy of Management Executive **18**(2): 6-20.

Gallivan, R. D. and G. Depledge (2003). "Trust, control and the role of interorganisational systems in electronic partnerships." Information Systems Journal **13**(2): 159-192.

Gammelgaard, J. (2010). "Knowledge retrieval through virtual communities of practice." Behaviour and Information Technology **29**(4): 349-362.

Gibson, C. B. and J. L. Gibbs (2006). "Unpacking the concept of virtuality: The effects of geographic dispersion, electronic dependence, dynamic structure and national diversity on team innovation." Administrative Science Quarterly **51**: 451-495.

Goldman, R. and R. Gabriel (2005). Innovation happens elsewhere : Open source as business strategy. USA, Elsevier Inc.

Gratton, L. and J. T. Erickson (2007). "Eight ways to build collaborative teams." Harvard Business Review: 102-109.

GridPP (2006a). Grid middleware report. http://www.gridpp.ac.uk/papers/.

GridPP (2006b). "GridPP: Development of the UK Computing Grid for particle physics." Journal of Physics G: Nuclear and Particle Physics **32**: N1-N20

GridPP. (2009). "LHC computing Grid ready for restart of the large hadron collider." http://www.gridpp.ac.uk/news/?p=97, Retrieved: 20th of October 2009.

Griffith, T. L., J. E. Sawyer, et al. (2003). "Virtualness and knowledge: Managing the love triangle of organizations, individuals, and information technology." MIS Quarterly **27**: 265-287.

Guba, E. G. and Y. S. Lincoln (1994). Competing paradigms in qualitative research. In: Handbook of qualitative research. N. K. Denzin and Y. S. Lincoln, Sage publications.

Guy, S. E. (2003). Patterns as artifacts in user-developer collaborative design. 8th European Conference of Computer-Supported Cooperative Work, Helsinki, Finland.

Hanseth, O. (1996). Information technology as infrastructure PhD, Goteborg University.

Hanseth, O. (2000). Infrastructures: From systems to infrastructures. In: Planet Internet. B. Dahlbom. Sweden, Lund.

Hanseth, O. and E. Monteiro. (1998). "Understanding information infrastructure." http://heim.ifi.uio.no/~oleha/Publications/bok.html Retrieved 15th of June, 2006.

Hansson, C., Y. Dittrich, et al. (2006). "How agile are industrial software development practices " Journal of Systems and Software **29**(9): 1295–1311.

Hardy, C., N. Phillips, et al. (2003). "Resources, knowledge and influence: The organizational effects of interorganizational collaboration." Journal of Management Studies **40**(2): 321-347.

Hasan, H. (1999). "Integrating IS and HCI using activity theory as a philosophical and theoretical basis." Australasian Journal of Information Systems (AJIS) **6**(2): 44-55.

Hasan, H. (2001). An overview of different techniques for applying activity theory to information systems. In: Information systems and activity theory: Theory and practice. H. Hasan, University of Wollongong Press.

Hasu, M. and Y. Engestrom (2000). "Measurement in action: An activity theoretical perspective on producer-user interaction." International Journal of Human Computer Studies **53**: 61-89.

Herbsleb, D. J. (2007). Global software engineering: The future of socio-technical coordination. ICSE International Conference on Software Engineering, IEEE Computer Society Washington, DC, USA.

Herbsleb, D. J. and A. Mockus (2003). "An empirical study of speed and communication in globally distributed software development." IEEE Transactions on Software Engineering **29**(6): 481-494.

Herbsleb, D. J., A. Mockus, et al. (2001). An empirical analysis of global software develoment: Distance and speed. 23rd International Conference on Software Enginnering (ICSE'01), IEEE Computer Society.

Herbsleb, D. J. and D. Moitra (2001). "Global software development." IEEE Software.

Herbsleb, D. J., J. D. Paulish, et al. (2005). Global software development at Siemens: Experience from nine projects. International Conference on Software Engineering (ICSE'05). St. Louis, Missouri, USA, ACM.

Herbsleb, J. D., D. L. Atkins, et al. (2002). Introducing instant messaging and chat into the workplace. Conference on Computer Human Interaction, Minneapolis.

Hey, T. and A. Trefethen (2002). "The UK e-Science core programme and the Grid." Future Generation Computer Systems **18** 1017-1031.

Hey, T. and A. Trefethen (2005). "Cyberinfrastructure for e-Science." Science **308**(5723): 817-821

Hey, T. and A. Trefethen (2008). E-Science, cyberinfrastructure and scholarly communication. In: Scientific collaboration on the internet. G. M. Olson, A. Zimmerman and N. Bos. Cambridge, MA:MIT Press.

Hey, T., S. Tansley, et al. (Eds). (2009). The fourth paradigm: Data-intensive scientific discovery. Redmond, Washington, Microsoft Research.

Highsmith, J. (2003). Agile project management: Principles and tools. Cutter Consortium. Arlington, MA.

Highsmith, J. and A. Cockburn (2001). "Agile software development: The business of innovation." IEEE Computer Society.

Hildreth, P., C. Kimble, et al. (1998). Computer mediated communications and international communities of practice. Ethicomp'98 Conference, The Netherlands.

Hildreth, P., C. Kimble, et al. (2000). "Communities of practice in the distributed international environment." Journal of Knowledge Management **4**(1): 27 - 38.

Hinds, P. J. and M. Mortensen (2005). "Understanding conflict in geographically distributed teams: The moderating effects of shared identity, shared context, and spontaneous communication." Organization Science **16**(3): 290-307.

Hirschheim, R. A. (1985). Information systems epistemology: A historical perspective. In: Information systems research: Issues, methods and practical guidelines. R. D. Galliers, Alfred Waller, Henley-on-Thames.

Huang, H. and E. Trauth (2007). Cultural influences and globally distributed information systems development: Experiences from Chinese IT professionals. Special Interest Group on Computer Personnel Research Annual Conference (SIGMIS-CPR'07), St. Louis, Missouri, USA ACM Press

Huang, J., S. Newell, et al. (2003). Conceptualizing the three dimensions of inter-organizational communities of practice. 11th European Conference on Information Systems, Naples, Italy.

Hughes, J. and T. Wood-Harper (1999). "Systems development as a research act." Journal of Information Technology **14**: 83-94.

Huitt, W. (2003). "Constructivism." Educational Psychology Interactive.
Hyssalo, S. (2005). "Objects and motives in a product design process." Mind, Culture, and Activity **12**: 19-38.
Hyysalo, S. and J. Lehenkari (2001). An activity-theoretical method for studying dynamics of user-participation in IS design. 24th Information Systems Research Seminar in Scandinavia (IRIS 24), Norway.
Iacovou, C., I. Benbasat, et al. (1995). "Electronic data interchange and small organisations: Adoption and impact of technology." MIS Quarterly **19**(4): 465-485.
Jap, S. D. and J. J. Mohr (2002). "Leveraging internet technologies in B2B relationships." California Management Review **44**: 24-38.
Jarzabkowski, P. (2003). "Strategic practices: An activity theory perspective on continuity and change." Journal of Management Studies **40**(1): 23-55.
Jasperson, J., P. E. Carter, et al. (2005). "A comprehensive conceptualisation of post-adoptive behaviour associated with information technology enabled work systems." MIS Quarterly **29**(3): 525-557.
Javenpaa, S. L., K. Knoll, et al. (1998). "Is anybody out there? Antecedents of trust in global virtual teams." Journal of Management Information Systems **14**(4): 29-64.
Javenpaa, S. L. and D. E. Leidner (1999). "Communication and trust in global virtual teams." Organization Science **10**(6): 791-815.
Jensen, C. and W. Scacchi (2005). Collaboration, leadership, control and conflict negotiation in the netbeans.org open source software development community. 38th Annual Hawaii International Conference on System Sciences (HICSS'05).
Jha, S. and M. Watson-Manheim (2007). Conduct, performance, & dilemmas of inter-organizational virtual organizing: A literature review. Virtuality and Virtualization: The International Federation for Information Processing (IFIP) Working Group 8.2 on Information Systems and Organizations and 9.5 on Virtuality and Society. S. S. Wanda Orlikowski, Kevin Crowston, Eleanor Wynn, Cynthia K. Pickering.
Jirotka, M., R. Procter, et al. (2005). "Collaboration and trust in healthcare innovation: The eDiamond case study." Computer Supported Cooperative Work **14**: 369-398.
John-Steiner, V., R. J. Weber, et al. (1998). "The challenge of studying collaboration." American Educational Research Journal **35**(4): 773-783.
Jonassen, D. H. and L. Rohrer-Murphy (1999). "Activity theory as a framework for designing constructivist learning environments." Educational Technology, Research & Development **47**(1): 61-79.
Jorgensen, N. (2001). "Putting it all in the trunk: Incremental software development in the freeBSD open source project." Information Systems Journal **11**(4): 321-336.
Kallinikos, J. (1999). "Computer-based technology and the constitution of work: A Study on the Cognitive Foundations of Work." Accounting, Management and Information Technologies **9**(4): 261-291.
Kaplan, B. and J. A. Maxwell (1994). Qualitative research methods for evaluating computer information systems. In: Evaluating health care information systems: Methods and applications. G. J. Anderson, E. C. Aydin and J. S. Jay, Sage, Thousand Oaks, CA**:** 45-68.

Kaptelinin, V. (1996). Activity theory: Implications for human-computer interaction. In: Context and Consciousness: Activity theory and human computer interaction. B. Nardi. Cambridge, MIT Press.

Kaptelinin, V. and B. Nardi (2006). Acting with technology: Activity theory and interaction Design. Cambridge, MIT Press.

Kaptelinin, V., B. Nardi, et al. (1999). "The activity checklist: A tool for representing the “space” of context." Interactions **6**(4).

Kasper-Fuehrer, E. and N. M. Ashkanasy (2001). "Communicating trustworthiness and building trust in interorganizational virtual organizations." Journal of Management **27**: 235-254.

Keen, P. G. W. (1991). Shaping the future: Business design through information technology. Boston, Harvard Business School Press.

Kelsey, D. (2010). Security policy group – SPG. Proceedings of the EGI Technical Forum, Amsterdam.

Kettunen, P. and M. Laanti (2007). "Combining agile software projects and large-scale organisational agility." Software Process: Improvement and Practice **13**(2): 183-193.

Kimble, C. and P. Hildreth (2004). Communities of practice: Going one step too far? 9th Colloque de L'AIM Conference, Evry, France.

Kimble, C. and P. Hildreth (2005). "Dualities, distributed communities of practice and knowledge management." Journal of Knowledge Management **9**(4): 102-113.

Klein, H. K. and M. D. Myers (1999). "A set of principles for conducting and evaluating interpretive field studies in information systems." MIS Quarterly **23**(1): 67-93.

Knorr-Cetina, K. (1999). Epistemic cultures: How the sciences make knowledge. Cambridge MA, Harvard University Press.

Knorr-Cetina, K. (2001). Objectual practice. In: The practice turn in contemporary theory. T. R. Schatzki, K. Knorr-Cetina and E. Von Savigny. London, Routledge.

Korpela, M., A. Mursu, et al. (2002). "Information systems development as an activity." Computer Supported Cooperative Work **11**(111-128).

Korpela, M., K. C. Olufokunbi, et al. (1998). Activity analysis as a method for information systems development: General introduction and experiments from Nigeria and Finland. ISCRAT'98 conference.

Korpela, M., H. A. Soriyan, et al. (2000). "Activity analysis as a method for information systems development." Scandinavian Journal of Information Systems **12**.

Kotlarsky, J., I. Oshri, et al. (2008). Globally distributed component-based software development: An exploratory study of knowledge management and work division. In: Knowledge processes in globally distributed contexts J. Kotlarsky, I. Oshri and P. Van Fenema. Hampshire, Palgrave MacMillan.

Kotlarsky, J. and P. Van Fenema (2008). Bridging gaps in globally dispersed collaboration: Developing knowledge bases. In: Knowledge processes in globally distributed contexts J. Kotlarsky, I. Oshri and P. Van Fenema. Hampshire, Palgrave MacMillan.

Kotlarsky, J., P. Van Fenema, et al. (2008). Developing a knowledge based perspective on coordination: The case of global software development projects. In: Knowledge processes in globally distributed contexts. J. Kotlarsky, I. Oshri and P. Van Fenema. Hampshire, Palgrave MacMillan.

Kussmaul, C., R. Jack, et al. (2004). Outsourcing and offshoring with agility: A case study. 4th Conference on Extreme Programming and Agile Methods, Canada, Springer.

Kuutti, K. (1991). Activity theory and its applications to information systems research and development. Information Systems Research: Contemporary Approaches & Emergent Traditions. H. E. Nissen, H. K. Klein and R. Hirschheim, Amsterdam: North-Holland**:** 529-550.

Kuutti, K. (1992b). Identifying potential CSCW applications by means of activity theory concepts: A case example. Conference of Computer Supported Cooperative Work, Toronto, Canada.

Kuutti, K. (1996). Activity theory as a potential framework for human-computer interaction research. In: Context and Consciousness: Activity theory and human computer interaction. B. Nardi. Cambridge, MIT Publisher**:** 17-44.

Kyriakidou, A. and W. Venters (2007). The multi-disciplinary development of collaborative Grids: The social shaping of a Grid for healthcare **(nominated for best paper)** 15th European Conference on Information Systems (ECIS'07), Switzerland.

Kyriakidou, A. and W. Venters (2009). Distributed large-scale systems development: Exploring the collaborative development of the particle physics Grid. 5th International Conference on e-Social Science, Cologne.

Lacity, M. C. and L. P. Willcocks (2001). Global information technology outsourcing: In search of business advantage, John Wiley& Sons Ltd.

Lanzara, G. F. (1999). "Between transient constructs and persistent structures: Designing systems in action." Journal of Strategic Information Systems **8**: 331-349.

Latour, B. (2005). Reassembling the social: An introduction to actor network theory. New York, Oxford University Press.

Lave, J. and E. Wenger (1991). Situated learning: Legitimate Peripheral participation. New York, Cambridge university press.

Layman, L., L. Williams, et al. (2006). "Essential communication practices for extreme programming in a global software development team." Information and Software Technology **48**: 781-794.

Lee-Kelley, L., A. Crossman, et al. (2004). "A social interaction approach to managing the 'invisibles' of virtual teams." Industrial Management and Data Systems **104**(8): 650-657.

Lee, A. S. and R. L. Baskerville (2003). "Generalizing generalizibility in information systems research." Information Systems Research **14**(3): 221-243.

Lee, A. S. and G. S. Hubona (2009). "A scientific basis for rigor in information systems research." MIS Quarterly **33**(2): 237-262.

Lee, G., W. Delone, et al. (2006). "Ambidextrous coping strategies in globally distributed software development projects." Communications of the ACM **49**(10).

Leffingwell, D. (2007). Scaling software agility: Best practices for large enterprises, Pearson education, Inc.

Lehenkari, J. and S. Hyysalo (2003). Activity theory in understanding distributed design and use of technology. 8th European Conference of Computer Supported Cooperative Work (ECSCW), Helsinki, Finland.

Leont'ev, A. N. (1978). Activity, consiousness and personality, Prentice Hall, Englewood Cliffs, NJ.

Leont'ev, A. N. (1981). Problems of the development of the mind. Moscow, Progress publishing.

Lero, X. W. and K. Conboy (2009). Understanding agility in software development from a complex adaptive systems perspective. 17th European Conference on Information Systems, Verona, Italy.

LHC accelerator (2011) [image online]. Available at: http://news.discovery.com/space/lhc-cern-new-collision-record-higgs-110523.html. Retrieved May 2011

LHC ring (2011) [image online]. Available at: http://scienceblogs.com/startswithabang/2011/05/the_future_of_colliders_beyond.php. Retrieved May 2011

Lindvall, M., V. Basili, et al. (2002). Empirical findings in agile methods. 2nd XP Universe and 1st Agile Universe Conference on Extreme Programming and Agile Methods.

Lings, B., B. Lundell, et al. (2006). Ten strategies for succesful distributed development. IFIP WG8.6 Working Conference: The transfer and diffusion of IT for organisational resilience, National University of Ireland, Galway.

Lloyd, S. (2003). "Building the next IT revolution." Physics World.

Lloyd, S. (2006). From Web to the Grid. PPARC and Parliament and House magazine.

Lui, T. W. and G. Piccoli (2007). Degrees of agility: Implications for information system design and firm strategy. In:Agile information systems: Conceptualisation, construction and management. K. C. DeSouza. Burlington, MA, Elsevier**:** 122-133.

Lyytinen, K. and G. M. Rose (2006). "Information system development agility as organizational learning." European Journal of Information Systems **15** (2).

Madey, G., V. Freeh, et al. (2002). The open source software development phenomenon: An analysis based on social network theory. 8th American Conference on Information Systems (AMCIS), Dallas.

Mahoney, J. M. (2004). "What is constructivism and why is it growing?" Contemporary Psychology **49**: 360-363.

Majchrzak, A., R. E. Rice, et al. (2000). "Computer-mediated inter-organizational knowledge-sharing: Insights from a virtual team innovating using a collaborative tool." Information Resources Management Journal **13**(1): 44-54.

Mannix, E. A., T. L. Griffith, et al. (2002). The phenomenology of conflict in virtual work teams. In: Distributed work. P. J. Hinds and S. Kiesler. Cambridge, MA, MIT Press.

Mattessich, P., M. Murray-Close, et al. (2001). Collaboration: What makes it work. St Paul, Minnesota, Amherst H. Wilder Foundation.

Maznevski, M. L. and K. M. Chudoba (2000). "Bridging space over time: Global virtual team dynamics and effectiveness." Organisation Science **11**(5): 473-492.

McLure Wasko, M. and S. Faraj (2000). " It is what one does: Why people participate and help others in electronic communities of practice." Journal of Strategic Information Systems **9**(2-3): 155-173.

McLure Wasko, M. and S. Faraj (2005). " Why should I share? Examining social capital and knowledge contribution in electronic networks of practice." MIS Quarterly **29**(1): 35-38.

Miettinen, R. and M. Hasu (2002). "Articulating user needs in collaborative design: Towards an activity theoretical approach." Computer Supported Cooperative Work **11**(1-2): 129-151.

Mockus, A. and D. M. Weiss (2001). "Globalization by chunking: A quantitative approach." IEEE Software **30**(7).

Mursu, A., I. Luukkomen, et al. (2007). "Activity theory in information systems research and practice: Theoretical underpinnings for an information systems development model." Information Research **12**(3).

Mwanza, D. (2001). Where theory meets practice: A case for an activity theory based methodology to guide computer system design. INTERACT' 2001: 8th IFIP conference on Human-Computer Interaction, Tokyo, Japan.

Mwanza, D. and Y. Engestrom (2005). "Managing content in e-learning environments." British Journal of Educational Technology **36**(3): 453-463.

Myers, D. M. (1997). "Qualitative research in information systems." MISQ Discovery.

Myers, D. M. and A. David (2002). An introduction to qualitative research in IS. In: Qualitative research in information systems. D. M. Myers and A. David. London, SAGE publications Ltd.

Nandhakumar, J. and D. Avison (1999). "The fiction of methodological development: A field study of information systems development." Information Technology & People **12**(2): 176-191.

Nandhakumar, J. and R. L. Baskerville (2006). "Durability of online teamworking: Patterns of trust." Information Technology and People **19**(4): 371-389.

Nandhakumar, J. and M. Jones (1997). "Too close for comfort? Distance and engagement in interpretive information systems research." Information Systems Journal **7**(2): 109-131.

Nardi, B. (1996). Context and consciousness: Activity theory and human-computer interaction. Cambridge, MIT Publisher.

Nardi, B. (1996a). Activity theory and human-computer interaction. In:Context and consciousness: Activity theory and human-computer interaction. B. Nardi. Cambridge, MIT Publisher.

Nardi, B. and J. R. Miller (1990). An ethnographic study of distributed problem solving in spread-sheet development. Conference of Computer Supported Cooperative work (CSCW'90), ACM Press.

Nardi, B. and J. R. Miller (1991). "Twinkling lights and nested loops: Distributed problem solving and spreadsheet development." International Journal of Man-Machine Studies **34**: 161-184.

Nentwich, M. (2008). Cyberscience: The age of digitilized collaboration? In: Scientific collaboration on the internet. . G. M. Olson, A. Zimmerman and N. Bos. Cambridge, MA:MIT Press.

Nerur, S., R. Mahapatra, et al. (2005). "Challenges of migrating to agile methodologies." Communications of the ACM **48**(5).

Ngwenyama, O. (2002). The critical social theory approach to information systems problems and challenges. In: Qualitative research in information systems. M. D. Myers and D. Avison, Sage Publications.

Nielsen, K. (2003). The development of collective competencies in the context of new high tech ventures. 5th International Conference in Organisational Learning and Knowledge.

Nohria, N. (1995). Note on organization structure. Reprint no 9-491-083, Harvard Business School.

Nørbjerg, J. and P. Kraft (2002). Software practice is social practice. In: Social thinking, software practice. Y. Dittrich, C. Floyd and R. Klischewski.

Novosti, R. (2009) "World's largest atom smasher back to life after power cut." Science.

O'Leary, M., W. Orlikowski, et al. (2002). Distributed work over the centuries: Trust and control in the hudson's bay company. In: Distributed Work. P. Hinds and S. Kiesler. Cambridge Ma: MIT Press., Ma: MIT Press.

Olson, G. M. and J. S. Olson (2000). "Distance matters." Human Computer Interaction **15**: 139-179.

Olson, G. M., A. Zimmerman, et al. (2008). Scienctific collaboration on the internet. Cambridge, MA: MIT Press.

One-Ki, L., B. Probir, et al. (2006). "Aligning IT components to achieve agility in globally distributed systems development." Communications of the ACM **49**(10).

Orlikowski, W. J. (1992). "The duality of technology: Rethinking the concept of technology in organizations." Organization Science **3**(3): 398-427.

Orlikowski, W. J. (1996). "Improvizing organizational transformation over time: A situated change perspective." Information Systems Research **7**(1): 63-92.

Orlikowski, W. J. (2000). "Using technology and constituting structures: A practice lens for studying technology in organisations." Organisation Science **11**(4): 404-428.

Orlikowski, W. J. (2002). "Knowing in practice: Enacting collective capability in distributed organizing." Organization Science **13**(3): 249-273.

Orlikowski, W. J. and J. J. Baroudi (1991). "Studying IT in organizations: research approaches and assumptions." Information Systems Research **2**(1): 1-28.

Orlikowski, W. J. and J. J. Baroudi (2002). Studying information technology in organisations: Research approaches and assumptions. In: Qualitatice research in information systems: A reader. London, Sage publications. **51-77**.

Orlikowski, W. J. and D. C. Gash (1994). "Technological frames: Making sense of information technology in organizations." ACM Transactions on Information Systems **12**(2): 174-207.

Orlikowski, W. J. and J. Yates (2002). "It's about time: Temporal structuring in organisations." Organisation Science **13**(6): 684-700.

Orr, J. (1990). Sharing Knowledge, celebrating identity: War stories and community memory in a service culture. In: Collective remembering: Memory in society. D. S. Middleton and D. Edwards. Beverley Hills, CA: Sage Publications.

Oshri, I., J. Kotlarsky, et al. (2008). Socialization in global contexts: Lessons from dispersed teams. In: Exploring virtuality within and beyond organizations: Social global and local dimensions. N. Panteli and M. Chiasson. Hampshire, Palgrave MacMillan

Oshri, I., J. Kotlarsky, et al. (2008b). Expertise management in a distributed context: The case of offshore information technology outsourcing. In: Knowledge processes in globally distributed contexts. J. Kotlarsky, I. Oshri and P. Van Fenema. Hampshire, Palgrave MacMillan.

Oshri, I., J. Kotlarsky, et al. (2007). "Managing dispersed expertise in IT offshore outsourcing: Lessons from Tata consultancy services." MISQ Executive **6**(2): 53-65.

Oshri, I., P. Van Fenema, et al. (2008). Knowledge transfer in globally distributed teams: The role of transactive memory. In: Knowledge processes in globally distributed contexts. J. Kotlarsky, I. Oshri and P. Van Fenema. Hampshire, Palgrave MacMillan.

OSIgroup. (2006). "Report of the working group on virtual research communities." http://www.nesc.ac.uk/documents/OSI/vrc.pdf Retrieved 22/02/2007.

OSIgroup. (2007). "Developing the UK's e-infrastructure for science and innovation." http://www.nesc.ac.uk/documents/OSI/report.pdf Retrieved 22/02/2007.

Paasivaara, M. and C. Lassenius (2003). "Collaboration practices in global inter-organizational software development projects." Software Process Improvement and Practice **8**: 183-199.

Paasivaara, M. and C. Lassenius (2004). Using iterative and incremental process in global software development. Workshop on Global Software Development, International Conference on Software Engineering (ICSE 2004), Edinburgh, Scotland, IEEE.

Pant, S. and C. Hsu (1995). Strategic information systems planning: A review. Information Resources Management Association International Conference, Atlanta, Georgia.

Panteli, N. and M. Chiasson (2008). Rethinking virtuality. In: Exploring virtuality within and beyond organizations: Social global and local dimensions. N. Panteli and M. Chiasson. Hampshire, Palgrave MacMillan.

Panteli, N. and M. Chiasson (2008a). Exploring virtuality within and beyond organizations: Social global and local dimensions. Hampshire, Palgrave MacMillan.

Panteli, N. and S. Sockalingam (2005). "Trust and conflict within virtual inter-organizational alliances: A framework for facilitating knowledge sharing." Decision Support Systems **39**: 599-617.

Parnas, D. (2006). "Agile methods and global software development: The wrong solution to an old but real problem." Communications of the ACM **49**(10).

Pass, S. (2004). Parallel paths to constructivism: Jean Piaget and Lev Vygotsky. Connecticut, Information Age Publishing.

Pearce, S. and W. Venters (2011). How particle physicists constructed the world's largest Grid: A case study in participatory cultures. Routledge handbook of participatory cultures. D. Aaron and J. Henderson, Routledge.

Pearce, S. E. (2004). "An open source Grid for science." www.gridpp.ac.uk/papers/EU_Sarah_Pearce.pdf, 25 March 2010.

Pinch, T. and W. Bijker (1984). "The social construction of facts and artifacts: Or how the sociology of science and the sociology of technology might benefit each other." Social Studies of Science **14**: 399-441.

Porra, J. (1996). Colonial systems, information colonies and punctuated prototyping, University of Jyvaskyla.

Porra, J. (1999). "Colonial systems." Information Systems Research **10**(1): 38-69.

Porra, J. (2010). Group-level evolution and information systems: What can we learn from animal colonies in nature? In: Evolutionary psychology and information systems research. N. Kock, Springer Science+Business Media.

Porra, J. and M. S. Parks (2006). "Sustaining virtual communities: Suggestions from the colonial model." Information Systems and e-Business Management **4**: 309-341.

Powell, W. W., K. W. Koput, et al. (1996). "Interorganisational collaboration and the locus of innocation." Administrative Science Quarterly **41**(1): 116-145.

Prikladnicki, R., J. L. N. Audy, et al. (2003). "Global software development in practice - Lessons learned." Software Process Improvement and Practice **8**: 267-281.

Qureshi, S. and I. Zigurs (2001). "Paradoxes and prerogatives in global virtual collaboration." Communications of the ACM **44**(12).

Ramesh, B., L. Cao, et al. (2006). "Can distributed software development be agile?" Communications of the ACM **49**(10).

Rathswohl, E. J. (1991). Applying Don Ihde's phenomenology of instrumentation as a framework for designing research in information science. In: Information systems research: Contemporary approaches and emergent traditions. H. E. Nissen, H. K. Klein and R. A. Hirschheim. Amsterdam**:** 421-438.

Reckwitz, A. (2002). "Toward a theory of social practices: A development in culturalist theorizing." European Journal of Social Theory **5**(2): 243-263.

Repenning, A., A. Ioannidou, et al. (2001). "Using components for rapid distirbuted software development." IEEE Software **18**(2): 38-45.

Rieber, R. W. and J. Wollock (1997). Consciousness as a problem for the psychology of behaviour. In: The collected works of L.S. Vygotsky (Volume 3). R. W. Rieber and J. Wollock. New York, Plenum publishing coorporation.

Rios, A. S., F. Aguilera, et al. (2009). "Virtual communities of practice's purpose evolution analysis using a concept-based mining approach." Knowledge-Based and Intelligent Information and Engineering Systems: Lecture Notes in Computer Science **5712**: 480-489.

Robey, D. (1991). Designing organisations, Irwin, Burr Ridgw, IL.

Robey, D., G. Im, et al. (2008). "Theoretical foundation of empirical research on interorganisational systems: Assessing past contributions and guiding future directions." Journal of the Association for Information Systems **9**(9): 497-518.

Robey, D., H. Khoo, et al. (2000). "Situated learning in cross-functional virtual teams." IEEE Transactions on Professional Communications **43**(1): 51-66.

Robottom Reis, C. and R. De Mattos Fortes (2002). An overview of the software engineering process and tools in the mozilla project, Universidade de S˜ao Paulo and Async Open Source company.

Roth, W. M. (2007). "Emotion at work: A contribution to third-generation cultural-historical activity theory." Mind, Culture and Activity **14**(1-2): 40-63.

Rupino da Cuhna, P. and A. Dias de Figueiredo (2001). Information systems development as flowing wholeness. IFIP Conference. Boise, Idahom USA, Kluwer academic publishers.

Russo, N. and E. Stolterman (2000). "Exploring the assumptions underlying information systems methodologies: Their impact on past, present and future ISM research." Information Technology and People **13**(4): 313-327.

Sarker, S. and S. Sahay (2004). "Implications of space and time for distributed work: An interpretive study of US Norwegian systems developmen teams." European Journal of Information Systems **13**: 3-20.

Sarker, S. and S. Sarker (2009). "Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context." Information Systems Research **20**(3): 440-461.

Sauer, J. (2006). Agile practices in offshore outsourcing - An analysis of published experiences. 29th Information Systems Research Seminar, Scandinavia.

Scacchi, W. (2002). Is open source software development faster, better and cheaper than software engineering? 2nd ICSE Workshop on Open Source Software Engineering Orlando.

Scacchi, W. (2004a). "Free and open source development practices in the game community." IEEE Software **21**(1): 59-66.

Scacchi, W., J. Feller, et al. (2006). "Understanding free/open source software development processes." Software Process Improvement and Practice **11**: 95-105.

Schatzki, T. R. (2001). Practice mid-ed orders. In: The practice turn in contemporary theory. R. T. Schatzki, K. Knorr-Cetina and E. Von Savigny.

Schatzki, T. R., K. Knorr-Cetina, et al. (2001). The practice turn in contemporary theory. London, Routledge.

Schmidt, C., B. K. Temple, et al. (2008). Virtuality in organizational team environments: Concept, measurement, and effects in the context of sustainable management. In: Exploring virtuality within and beyond organizations: Social global and local dimensions. N. Panteli and M. Chiasson. Hampshire, Palgrave MacMillan.

Schultze, U. and W. J. Orlikowski (2001). "Metaphors of virtuality: shaping an emergent reality." Information and Organization **11**: 45-77.

Schwen, T. M. and N. Hara (2003). "Community of practice: A metaphor for online design." The Information Society **19**: 257-270.

Scott, S. V. (2002). Lived methodology: A situated discussion of truth and method in interpretive information systems research. London, London school of economics and political science.

Sengupta, B., S. Chandra, et al. (2006). A research agenda for distributed software development. 28th International Conference on Software Engineering Shanghai, China ACM Press

Shrum, W., J. Genuth, et al. (2007). Structures of scientific collaboration. Cambridge, MA, MIT Press.

Simons, M. (2006). "Global software development: A hard problem requiring a host of solutions." Communications of the ACM **49**(10).

Smarr, L. (2004). Grids in context. In: The Grid2: Blueprint for a new computer infrastructure. I. Foster and C. Kesselman, Elsevier Inc.

Smite, D. and J. Borzovs (2006). A framework for overcoming supplier related threats in global Projects. 13th European Conference on Software Process Improvement (EuroSPI ' 06), Joensuu, Finland.

Smite, D., C. Wohlin, et al. (2010). "Empirical evidence in global software enginnering: A systematic review." Empirical Software Engineering **15**: 91-118.

Smith, P. G. and E. L. Blanck (2002). "From experience: Leading dispersed teams." The Journal of Product Innovation Management **19**(4): 294-304.

Smits, H. (2007). The impact of scaling on planning activities in an agile software development context. 40th Hawaii International Conference on System Sciences.

Spasser, M. A. (1999). "Informing information science: The case for activity theory." Journal of the American Society for Information Science and Technology **50**(12): 1136-1138.

Stake, R. E. (1995). The art of case study research. US, Sage publications.

Star, S. and K. Ruhleder (1996). "Steps towards an ecology infrastructure: Borderlines of design and access for large information spaces." Information Systems Research **7**(1): 111-134.

Sutherland, J., A. Viktorov, et al. (2007). Distributed scrum: Agile project management with outsourced development teams. 40th Hawaii International Conference on System Sciences.

Swan, J., H. Scarbrough, et al. (2002). "The construction of 'communities of practice' in the management of innovation." Management Learning **33**(4): 479-496.

Swanson, E. B. and N. C. Ramiller (1997). "The organizing vision in information systems innovation." Organization Science **8**(5): 458-474.

Swidler, A. (2001). What anchors cultural practices. In: The practice turn in contemporary theory. T. R. Schatzki, K. Knorr-Cetina and E. Von Savigny.

Tanner, M. (2009). "Communication and culture in global software development: The case of Mauritius and South Africa." Journal of Information, Information Technology, and Organizations **4**: 57-85.

Teigland, R. (2000). Communities of practice in a high-technology firm: The impact of internal and external sources of knowledge on individual performance In: The flexible firm 126 - 146

Teo, H. H., C. Ranganathan, et al. (2006). "Key dimensions of inhibitors for the deployment of web-based business-to-business electronic commerce." Engineering management, IEEE transactions **53**: 395-411.

Teo, H. H., K. K. Wei, et al. (2003). "Predicting intention to adopt interorganisational linkages: An institutional perspective." Management Information Systems **27**(1): 19-50.

Theunissen, M., A. Boake, et al. (2005). Open source and agile software development in corporates: A contradiction or an opportunity? Jacquard International Conference, Holland.

Thévenot, L. (2001). Pragmatic regimes governing the engagement with the world. In: The practice turn in contemporary theory. T. R. Schatzki, K. Knorr-Cetina and E. Von Savigny.

Tim-Berners, L. (1996). "The Web: Past, present and future." IEEE Computer

Trauth, E. M. and L. M. Jessup (2000). "Understanding computer-mediated discussions: Positivist and interpretive analyses of group support system use." MIS Quarterly **23**: 43-79.

Traweek, S. (1988). Beamtimes and lifetimes: The world of high energy physics. London, Harvard University press.

Truex, D., R. L. Baskerville, et al. (2000). "Amethodical systems development: The deferred meaning of systems development methods." Accounting, Management and Information technology **10**: 53-79.

Turner, P. and S. Turner (2001). "Describing team work with activity theory." Cognition, Technology and Work **3**: 127-139.

Turner, S. (1994). The social theory of practices: Tradition, tacit knowledge and presuppositions. Cigago, The University of Chigago Press.

Uden, L. (2007). "Activity theory for designing mobile learning." International Journal of Mobile Learning and Organisations **1**(1): 81-102.

Van Binsbergen, W. (1998). "Globalization and virtuality: Analytical problems posed by the contemporary transformaion of african societies." Development and Change **29**: 873-903.

Van Der Aa, O., S. Burke, et al. (2006). GridPP: Running a production Grid. UK e-Science All Hands Meeting Conference, Nottingham.

Van Dick, R., U. Wagner, et al. (2004). "The utility of a broader conceptualisation in organizational identification: Which aspects really matter?" Journal of Occupational and Organizational Psychology **77**: 171-191.

Van Fenema, P. (2002). Coordination and control of globally distirbuted software projects. PhD, Department of Decision and Information Sciences, Erasmus University, Rotterdam School of Management

Venters, W. and T. Cornford (2006). Introducing Pegasus: An ethnographic research project studying the use of Grid technologies by the UK particle physics community. 2nd International Conference on e-Social Science, Manchester, UK.

Venters, W. and B. Wood (2007). "Degenerative structures which inhibit communities of practice: A case study of the British Council." Information Systems Journal **17**: 349-368.

Venters, W., Y. Zheng, et al. (2007). Collaborative construction of Grid technology: Researching the usability of Grids within particle physics. 3rd International Conference on e-Social Science, Ann Arbor, Michigan US.

Vessey, I. and R. Glass (1998). "Strong vs weak approaches to systems development." Communications of the ACM **41**(4): 99-102.

Virkkunen, J. and K. Kuutti (2000). "Understanding organisational learning by focusing on activity systems." Accounting, Management and Information Technology **10**: 291-319.

Voss, A., M. Mascord, et al. (2007). e-Research infrastructure development and community engagement. UK e-Science All Hands Meeting, Nottingham.

Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. Harvard, Harvard university press.

Walsham, G. (1993). Interpreting information systems in organizations. Chichester, John Wiley & Sons.

Walsham, G. (1995). "The emergence of interpretativism in IS research." Information Systems Research **6**(4): 376-394.

Walsham, G. (2001). Making a world of difference: IT in a global context. Chichester, Whiley.

Walsham, G. (2006). "Doing interpretive research." European Journal of Information Systems **15**(3): 320-330.

Walsham, G. and C. Kwong Han (1993). "Information systems strategy formation and implementation: The case of a central government agency " Accounting, Management and Information Technologies **3**(3): 191-209

Walsham, G. and T. Waema (1994). "Information systems strategy and implementation: A case study of a building society." ACM Transactions on Information Systems **12**(2): 150-173.

Walther, J. B. and U. Bunz (2005). "The rules of virtual groups: Trust, liking and performance in computer-mediated communication." The Journal of Communication **55**(4): 828-846.

Ward, K. J. (1999). "The Cyber-ethnographic (re)construction of two feminist online communities." Sociological Research Online **4**(1).

Webster, F. (2005). "Making sense of the information age: Sociology and cultural studies." Information Communication and Society **8**(4): 439-458.

Weems-Landingham, V. (2008). Building virtual cooperation: Guidelines for effective performance. In: Exploring virtuality within and beyond organizations: Social global and local dimensions. N. Panteli and M. Chiasson. Hampshire, Palgrave MacMillan.

Weick, K. (1993). "The collapse of sensemaking in organizations: The mann gulch disaster." Administrative Science Quarterly **38**.

Weick, K. (1996). "Drop your tools: An allegory for organizational studies" Administrative Science Quarterly **41**: 301-313.
Weick, K. (1998). "Improvisation as a mindset for organizational analysis." Organization Science **9**(5): 543-555.
Weick, K. (1999). Sense making as an organisational dimension of global change. In: The human dimensions of global change. J. Dutton and D. Cooperrider, CA: Sage publications Inc.
Weick, K. (2001). Making sense of the organization, Blackwell Publishing.
Weick, K. (2002). The aesthetic of imperfection in orchestras and organisations. In: Organisational improvisation. K. N. Kamocher, C. M.P.E and D. C. J.V. New York, Routledge.
Weick, K. and H. K. Roberts (1993). "Collective mind in organizations: Heedful interrelating on flight decks." Administrative Science Quarterly **38**: 357-381.
Weick, K., K. M. Sutcliffe, et al. (1999). Organising for high reliability: Processes of collective mindfulness. In: Research in organizational behaviour. B. Staw and L. L. Cummings. Greenwich, Connecticut, JAI Press. **21**.
Weill, P. and M. Broadbent (1998). Leveraging the new infrastructure. Cambridge, Ma, Harvard business school press.
Wellman, B. and M. Gulia (1999). Net surfers don't ride alone. In: Networks in the global village: Life in contemporary communities. B. Wellman, Boulder, CO: Westview Press**:** 331-366.
Wenger, E. (1998). Communities of practice: Learning, meaning and identity. Cambrdige, United Kingdom, Cambridge University Press.
Wenger, E. and W. Snyder (2000). "Communities of practice: The organizational frontier". Harvard Business Review: 139-145.
Wenger, E., R. McDermott, et al. (2002). Cultivating communities of practice, Harvard Business School Press.
Whitehead, J. (2007). Collaboration in software engineering: A roadmap. ICSE International Conference on Software Engineering, IEEE Computer Society Washington, DC, USA
Wiesenfeld, B. M., S. Raghuram, et al. (1998). "Communication patterns as determinants of organizational identification in a virtual organization." Journal of Computer-Mediated Communications **3**(4).
Wiesenfield, B. M., S. Raghuram, et al. (2001). "Organizational identification among virtual workers: The role of need for affiliation and perceived work-based social support." Journal of Management **27**: 213-229.
Willcocks, L., W. Venters, et al. (2011). "Cloud and the future of business: From costs to innovation: Part 4: Innovation", Accenture. **4**.
Williams, L. and A. Cockburn (2003). "Agile software development: It's about feedback and change." IEEE **June**.
Wilson, T. D. (2006). "A re-examination of information seeking behaviour in the context of activity theory." Information Research **11**(4).
Wiredu, O. G. (2005). Mobile computing in work-integrated learning: Problems of remotely-distributed activities and technology use, London school of economics and political sciences.
Wladawsky-Berger, I. (2004). The industrial imperative. In: The Grid2: Blueprint for a new computing Infrastructure. I. Foster and C. Kesselman, Elsevier Inc.
Wouters, P. and P. Schroder (2003). Promise and practice in data sharing. Amsterdam, NIKI-KNAW.

Yalaho, A. (2006). A conceptual model of ICT-supported unified process of international outsourcing of software production. 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06), IEEE.

Yalaho, A., W. Chunling, et al. (2004). IT-supported international outsourcing of software production: An explorative case study. Information Resources Management Association (IRMA'04) Conference, New Orleans, USA, Idea Group Publishing.

Yan, L. (2008). Regulater and emergent identifications: The case of a virtual organization. In: Exploring virtuality within and beyond organizations: Social global and local dimensions. N. Panteli and M. Chiasson. Hampshire, Palgrave MacMillan.

Ye, Y. (2005). Dimensions and forms of knowledge collaboration in software development.12th Asia-Pacific software engineering conference (APSEC'05).

Yin, R. (2003). Case study research design and methods, Thousand Oaks: SAGE publications.

Yoo, Y. and P. Kanawattanachai (2001). "Developments of transactive memory systems and collective mind in virtual teams." International Journal of Organizational Analysis **9**(2): 187-208.

Zheng, Y., W. Venters, et al. (2007). Agility, improvisation, or enacted emergence. International Conference on Information Systems, Montreal, Canada.

Zheng, Y., W. Venters, et al. (2007b). Distributed development and scaled agility: Improvising a Grid for particle physics. London School of Economics Working Paper Series - 163.

Zhua, J., J. Gonga, et al. (2007). "A collaborative virtual geographic environment based on P2P and Grid technologies " Information Sciences **177**(21): 4621-4633.

# 11 Glossary and acronyms

| | |
|---|---|
| **ALICE** | One of the LHC experiments. |
| **AT** | Activity theory. The theoretical framework employed in this study. |
| **ATLAS** | One of the LHC experiments. |
| **CERN** | European organisation for nuclear research. It is based in Geneva and it is considered the world's large particle physics laboratory in the world. |
| **CoP** | Communities of practice. This study draws upon this concept, among others, to enrich AT, the theoretical framework employed here. It also helps define the HEAD_SDC, since this framework extends the work of such CoP in a global and virtual setting and identifies other "distinctive" characteristics which inform the work of such theories of community. |
| **CMS** | One of the LHC experiments. |
| **CPU** | Central processing unit. |
| **CSDAS** | Computer scientists developers sub-activity system. It is an instance of the overarching 'distributed systems development' activity system which interacts, competes and contradicts with the other two sub-activity systems (PPDAS and PPUDAS). |
| **CSGDC** | Computer scientists, members of the Grid development community. |
| **EDG** | European data Grid. Founded as the flagship EU project to develop a prototype Grid service. |
| **EGEE** | Enabling Grids for e-Science. An EU-funded project to develop a Grid infrastructure to support all sciences. EGEE closely works with LCG and GridPP for the Grid's development. |
| **EMT** | Engineering management team. Dealing with daily development issues from an EGEE perspective. |
| **ETICS** | e-Infrastructure for Testing, Integrational and Configuration of Software. A tool used for building software. |

**Experiments** The four LHC experiments, ATLAS, CMS, LHCb and ALICE. They are the users of the Grid infrastructure, providing their requirements to the developers.

**GDC** Grid development community. This research's focus and context. The Grid development community is an amalgam of the LCG, GridPP and EGEE projects and consists of both particle physicists and computer scientists members who are involved in the Grid's development, deployment and user support.

**GGUS** Global Grid User Support System.

**GridPP** Grid for particle physics. A collaboration of UK particle physics institutes, contributing to the development and deployment of the Grid for the wider particle physics community.

**GSD** Global software development.

**HEAD_SDC** Hybrid Experimental Agile Distributed Systems Development Communities. It is a framework of guidance for other organisations to reflect upon in order to engender and maintain such a community. This framework represents this study's major practical and theoretical contribution.

**HEP** High energy physics.

**ICT** Information communication technology.

**INFNGrid** Italy's National Institute for Nuclear Physics. INFNGrid is the Italian project contributing to LCG.

**IS** Information systems.

**ISD** Information systems development.

**IT** Information technology.

**LCG** Large Hadron Collider computing Grid. It is the project initiated by particle physics institutions around the world in order to gather the manpower, tools and funding to develop the world's largest Grid to support their LHC experimental research.

**LHC** Large Hadron Collider. A particle physics particle accelerator.

**LHCb** One of the LHC experiments.

| | |
|---|---|
| **Middleware** | Low-level software that enables the fabric (computers, storage and networks) to intercommunicate and allows the sharing of these resources via common Grid protocols. |
| **MOU** | Memorandum of understanding. It serves as a "gentleman's" agreement defining the degree of contribution of each participant institute to the LCG project. |
| **NoP** | Networks of practice. A term defined for a loosely knit, geographically distributed group of individuals engaged in shared practice. |
| **OSS** | Open source software. |
| **OSSD** | Open source software development. |
| **PP** | Particle physics. |
| **PPDAS** | Particle physics developers sub-activity system. It is an instance of the overarching 'distributed systems development' activity system which interacts, competes and contradicts with the other two sub-activity systems (CSDAS and PPUDAS). |
| **PPGDC** | Particle physicists (both employed to be developers and "users" acting as developers), members of the Grid development community. |
| **PPS** | Pre-production system. Runs like a mini-Grid where software can be tested before it moves into production. |
| **PPUDAS** | Particle physics users developers sub-activity system. It is an instance of the overarching 'distributed systems development' activity system which interacts, competes and contradicts with the other two sub-activity systems (CSDAS and PPUDAS). |
| **Production** | A term that particle physicists use for a working system. If the system is of production quality then it means it is working and can be deployed and used by users. |
| **RAL** | Rutherford Appleton Laboratory. RAL acts as the GridPP's Tier-1 contributing to Grid development and deployment. |
| **RGMA** | Relational Grid monitoring architecture. A Grid middleware component. |

| | |
|---|---|
| **Sites** | Administrative domains (IT centres) providing computing resources to the Grid. Sites are also responsible to provide user support. |
| **TCG** | Technical coordination group. Responsible for prioritising the requirements of the different communities involved in EGEE. |
| **VO** | Virtual organisation. A Grid’s VO is defined as a group of both users and computing resources from a number of real global organisations which are brought together to collaborate on a particular project in order to enable such sharing of geographically distributed resources. |
| **WLCG** | Worldwide Large Hadron Collider computing Grid collaboration. |
| **XP** | Extreme programming |

## Appendix A – Comparison of systems development practices

This Appendix provides a comparison between agile practices, open source development practices and the systems development practices identified in this case.

| | **Amethodical agile development practices** | **Open source development practices** | **Particle physics development practices** |
|---|---|---|---|
| **Reasons** | | | |
| Reaction against heavyweight methods. | Yes | Yes | Not really. They do not really care about what's out there. |
| Lack of experience. | Yes | Yes | Yes. The Grid is a new large-scale distributed technology, new experiments. The Grid is very complicated, it sits on many machines. They do not know what to expect. |
| Time pressure – rush to market. | Yes | Yes | Yes, very tight deadline of LHC. |
| Turbulent business environment. | Yes | Yes | Yes, overly complex and uncertain. Distribution and scale of the project and of the Grid is also a problem. These distinguish LCG from other open source or agile projects. |
| Deliver a piece of software | Yes | Yes | It is not fundamentally about delivering a piece of software or a system, but it is about doing new physics. |
| Globalisation. | No | Yes, distributed community of individual developers. | Yes, widely distributed community of many participants and institutes. The scale and the nature of the collaboration is different. |
| Vague requirements. | Yes | Yes | Yes. New technology, new experiments. |
| Existing culture influencing work. | Not really. Agile teams dissemble when the project finishes. No community behind it. | Not really. Independent individual contributions. | The culture and history of particle physicists are "distinctive" and influence the work. Pragmatism, trial and error, experimentation, improvisation. |
| **System development practices** | | | |
| Short iterative cycles of development. | Yes | Yes. Iterative development, incremental releases. | Yes, rapid development, incremental release cycle. |
| Parallel development. | Yes | Yes | Yes. Simultaneous or overlapping activities of development such as design, |

| | | | |
|---|---|---|---|
| | | | coding, testing, maintenance. |
| Rapid feedback. | Yes | Yes | Yes, the goal is immediate feedback. |
| Release orientation. | Yes | Yes. Viable release strategy is determined, such as mirroring and versioning since quality releases must be produced. | Yes, frequent releases, nightly builds. |
| Prototyping. | Yes | Yes | Yes, rapid prototyping. |
| Tool dependence. | Yes, but mostly off-the-shelf products. | Yes, but mostly off-the-shelf products. | Yes, but mostly in-house tools. |
| Customer involvement. | Yes | Developers are the customers. | Yes. Most of the developers are themselves the customers. The system is also developed with close involvement of the user community who exercise tremendous influence and pressures. Also, cultivation of "power users" to test the system from early on and support other users. |
| Coding your way out. | Yes | Yes | Yes, hacking is widely acceptable. |
| Standardised architecture. | Yes | Yes | No. It changes and evolves with user requirements. |
| Components based development. | Yes | No teams of developers exist. Rather individuals contributing code. Different nature of distributed development. Limited collaboration and face-to-face meetings are avoided. | Development work is structured as "clusters of competence". Different patches of expertise focusing on one middleware component but with strong collaboration between the different clusters. |
| Tailored methods. | Yes | Yes | Yes, Limited use of traditional methods. More reaction to what is happening, rather than employing certain methods. |
| Quality is negotiable. | Yes | Yes | Yes. But it still works. |
| Maintenance ignored. | Yes | No, but problematic. If core developers stop the development, and no other developers take up their tasks, the system quickly dies away. | Maintenance as evolutionary redevelopment and re-invention, rather than "maintenance" in the traditional sense. |
| Discourage | Yes | No, good | No, although codification of |

| | | | |
|---|---|---|---|
| documentation beyond the code. | | documentation is important. Separate documentation teams exist. | knowledge is problematic. However, wikis, websites, etc can be seen as places to find information/documentation about procedures, tools, etc. |
| Requirements gathering. | Frequent interaction between individuals. | Developers are usually themselves the users and therefore the requirements take less time to negotiate. | Interactive, informal process. Takes the form of threaded messages and discussions on wikis, quick feedback from prototypes. It is based on a combination between day-to-day requirements and preconceived ideas based on past experience (since most of the developers are the users). |
| Bug-driven development. | No | Yes. But systems are also developed based on modularity and module ownership. | Yes, bug driven and experimental development. |
| Competing parallel solutions. | No | No | Yes, that is how they work. The cream comes to the top, only the most relevant solutions survive. |
| Testing in a PPS. | No | No | Yes, pre-production serves as a mini-Grid service for testing packages before proceeding to production. |
| **Management strategy and collaboration** | | | |
| Planning. | Yes | Yes. Important, because face-to-face meetings are restricted. No central control though. | Limited, but vital to provide direction to the project. They know that nothing goes as planned. |
| Dependence on good people. | Yes | Yes. Even adding developers late may increase the functionality and quality of the system. | Yes. They value their people. Highly trustful environment. Intelligent and motivated people. Driven by shared goals and "sacred causes". |
| Rotation of team membership. | Yes | No, each member acquires power by sustained contributions over time. People take roles based on their skills, accomplishments, availability, etc. | Kind of. Rather, no fixed roles exist, individuals can do more than one job at a time. Voluntarism. |
| Leadership and collaboration style of management. | Yes | No leader, all members share equal power. Interlinked layered | No leadership. Democratic meritocracy. Very informal organizational structures. Their collaboration is driven |

| | | | |
|---|---|---|---|
| | | meritocracies operating as a dynamical organisation but loosely coupled virtual enterprise. | by “charismatic leadership”. Weak authority (management with “no teeth”). High autonomy and freedom. Not everybody shares equal power though. Respect to seniority. |
| Developers work in small teams with customers being active team members . | Yes | No, developers are distributed and they are usually the customers. | No because of distributed model of development. Collaboration of thousands of people, widely distributed around the world. End-users are active members. They are considered to be “powerful users”, highly dictating the work. |
| Self-organising teams. | Yes | Yes | Yes. Also, emerging power user communities to support the work, facilitate communication between developers and users and support other local users. |
| Collaborative but speedy decision-making process. | Yes | Yes | Yes. Democratic decision-making. Decisions are reached only when individual opinions merge into consensus. Minimal political influence. |
| Extensive communications. | Yes | Communication through emails. **But** the goal is to minimise collaboration and avoid face-to-face meetings. | Yes, overload of communications, meetings and emails. PP have “grand” meetings and value face-to-face interaction. Collaborating is “second-nature” to them. |
| Project is broken down into sub-projects, each of which usually involves planning, development, integration, testing, and delivery. | Yes | Yes | Not really, more structure and discipline is needed. |

This information is based on the research’s findings as well as on the following papers: (Truex, Baskerville et al. 2000; Cockburn and Highsmith 2001; Robottom Reis and De Mattos Fortes 2002; Erenkrantz and Taylor 2003; Baskerville and Pries-Heje 2004; Scacchi, Feller et al. 2006)

## Appendix B – Table of research activity

This Appendix provides an indicative outline of the key interviews, meetings and activities undertaken by the researcher. The table is a post-hoc selection of key events that provides a flavour of the research activity.

| Type of encounter | Job title/Role/ Team in attendance | Date | Location | Description |
|---|---|---|---|---|
| 17th GridPP collaboration meeting | GridPP PMB managers, developers (both particle physicists and computer scientists) system administrators, experimental users. | 1-2/11/2006 | Edinburgh | First encounter with the UK PP community. Discussions and presentations in the meeting revolved around middleware development and support, analysis on the Grid, Grid deployment, site installation and management, experiments' service challenges. |
| Interview | GridPP PMB member – also member of the Global Grid Forum. | 01/11/2006 | Edinburgh | |
| Informal discussion | GridPP members. | 01/11/2006 | Edinburgh | During dinner after the end of the meeting the researcher introduced her research. Informal discussion about GridPP and culture of physicists. |
| Informal discussion | GridPP middleware developers (both particle physicists and computer scientists). | 02/11/2006 | Edinburgh | During coffee after the end of the meeting. Discussion about Grid middleware, challenges, development methods, etc. |
| Interview | PhD particle physicist on ATLAS . | 06/12/2006 | UCL | |
| Interview | PhD student on CMS at Bristol. | 07/03/2007 | CERN | |
| Interview | Postdoc on CMS at Imperial. | 07/03/2007 | CERN | |
| Interview | IT Grid deployment. | 08/03/2007 | CERN | |
| Interview | Postdoc on CMS at Bristol. | 09/03/2007 | CERN | |
| Interview | EGEE head of | 09/03/2007 | CERN | |

| | | | | |
|---|---|---|---|---|
| | dissemination & communication. | | | |
| Interview | Grid deployment. | 12/03/2007 | CERN | |
| Participant observation | CMS user. | 18/05/2007 | Imperial | The user was observed while trying to submit jobs on the Grid. He guided the researcher through the Grid and showed what parts of the Grid he uses, which parts are problematic and what kind of applications have been developed to run on top of the Grid middleware. |
| Interview | Software developer. | 07/12/2007 | RAL | |
| Interview | Software developer – Manager of the RGMA team. | 07/12/2007 | RAL | |
| Interview | Software developer. | 07/12/2007 | RAL | |
| Interview | Software developer. | 07/12/2007 | RAL | |
| Interview | Software developer. | 15/02/2008 | RAL | |
| Interview | Software developer. | 15/02/2008 | RAL | |
| Interview | EGEE member –interpreting requirements and bringing back information to the UK. | 15/02/2008 | RAL | |
| Interview | Deployment group member. | 07/03/2008 | RAL | |
| Interview | LHCb computing coordinator – also PP applications developer. | 07/03/2008 | RAL | |
| Interview | Tier-2 manager and site admin. | 07/03/2008 | RAL | |
| 20th GridPP collaboration meeting | GridPP PMB managers, developers (CS and PP), system administrators, experimental | 11-12/03/2008 | Dublin | Discussions and presentations in the meeting revolved around introducing GridPP3, middleware storage element, deployment and |

| | | | | |
|---|---|---|---|---|
| | users. | | | support, security on the Grid, site updates, experiment updates as well as a thorough 1½ hour “users” discussion. |
| Informal discussion | System administrators, ATLAS and LHCb users. | 11/03/2008 | Dublin | During dinner after the meeting ended, the researcher had the opportunity to discuss about the work of systems administrators, the challenges they face, site problems, communication with developers and experimental physicists, etc., introduced her research. |
| Informal discussion | Developers (PP) and storage expert. | 11/03/2008 | Dublin | During coffee after the end of the meeting. Discussion about Grid middleware, challenges, development methods, communication with users and the emergence of the GridPP storage community to support users. |
| Interview | Tier-2 storage management support – also Tier-2 storage expert coordinating and communicating with system administrators at different sites and with developers. | 12/03/2008 | Dublin | |
| WLCG workshop | Global collaboration. LCG, GridPP and EGEE management board members, middleware developers (both computer scientists and particle physicists), experiments, | 21-25/04/2008 | CERN | People from all around the world involved in the collaboration attend the workshop. It is an opportunity for planning as well as obtaining information from CERN regarding what each country should do. Here, problems regarding experiments, operations, deployment, security, middleware |

| | | | | |
|---|---|---|---|---|
| | system administrators, deployers etc. | | | development, etc. are discussed. They also try to promote best practice techniques in software development in order to be able to run reliable services. They share expertise, information and knowledge. |
| Informal Discussion | LCG MB members, EGEE managers. | 21/04/2008 | CERN | Discussion over lunch. Challenges between EGEE and LCG, developing a Grid for HEP versus a Grid for all the sciences. Middleware competing solutions, e.g. the data management component, discussion about the middleware development cycle, etc. |
| Interview | Middleware site of the EDG – deputy of the TCG CERN - involved in EGEE project execution board. | 22/04/2008 | CERN | |
| Interview | LCG deployment. | 23/04/2008 | CERN | |
| Interview | PPS member. | 23/04/2008 | CERN | |
| Interview | Responsible for requirements gathering for storage SRM protocol. | 23/04/2008 | CERN | |
| Interview | Particle physicist. EGEE managerial site of middleware. Also LCG Grid deployment. | 23/04/2008 | CERN | |
| Interview | Data management clustering – middleware developer. | 24/04/2008 | CERN | |
| Interview | Working on the EGEE quality project – quality of information that goes with the | 24/04/2008 | CERN | |

| | | | | |
|---|---|---|---|---|
| | project, responsible for communication. | | | |
| Interview | Particle physicist, involved in data management development – now he is in operations. | 24/04/2008 | CERN | |
| Interview | Particle physicist. Middleware developer (on storage). | 25/04/2008 | CERN | |
| Interview | Particle physicist. Middleware architect – now working on ALICE experiment. | 25/04/2008 | CERN | |
| Interview | Cluster leader in Bologna (software developer) – now EGEE manager. | 25/04/2008 | CERN | |
| Interview | System administrator. | 25/04/2008 | CERN | |
| Participant observation | Particle physicist developer (storage component). | 26/04/2008 | CERN | Observed while writing code in C++, responding to a user's query, uploading comments on wiki about storage issues and while discussing with a colleague down the corridor about storage standards. |
| Participant observation | Particle physicist user – Alice experiment. | 27/04/2008 | CERN | Observed while trying to submit jobs on the Grid. He explained how doing analysis on the Grid works. The software running on top of the Grid middleware was not working properly, he discussed this with a colleague. His job was not running properly and he blamed the Grid middleware. He gave the researcher a tour of the ALICE experiment. |

| | | | | |
|---|---|---|---|---|
| Interview | Tier-1 manager. | 27/05/2008 | RAL | |
| Interview | PPS member. | 27/05/2008 | RAL | |
| Interview | Deployment team - specialises in computing element and storage. | 27/05/2008 | RAL | |
| Interview | Deployment team – specialises in storage. | 27/05/2008 | RAL | |
| Participant observation | Software developer on the RGMA component. | 27/05/2008 | RAL | Observed while redesigning the RGMA component. He was closely collaborating with colleagues who worked down the corridor. The RGMA team manager was providing directions on how to proceed. |
| UK e-Science hands-on meeting | Scientists from various scientific domains (e.g. physics, astronomy, chemistry, etc.) and from different projects (e.g. ATLAS project, CMS project, LHCb project, GridPP project, AstroGrid etc.). | 8-11/09/2008 | Edinburgh | Opportunity to share knowledge, ideas and findings within various scientific domains. Each project is trying to promote their research. You could clearly see the strong identity of GridPP and LHC experiments through the logoed pens and bags, logoed T-shirts, etc. Presentations of papers, workshops and poster session. Lots of technical discussions and sharing of expertise around Grid middleware development. |
| 22nd GridPP collaboration meeting | GridPP PMB managers, developers (CS and PP) system administrators, experimental users. | 1-2/04/2009 | UCL | Discussions revolved around GridPP3 updates, project management, site updates, experiments problems, etc. |
| Informal discussion | GridPP PMB member. Storage middleware | 01/04/2009 | UCL | During coffee after the end of the meeting. Discussions about GridPP, LCG and |

| | | | | |
|---|---|---|---|---|
| | developer. | | | EGEE. Discussions about the culture of particle physicists and this nature of volunteering and working through competing solutions. Updates on the Grid and its usability. |
| Informal discussion | Technical coordinator at Imperial. | 01/04/2009 | UCL | Discussion about system administrators, problems different sites face, etc. |
| Discussion over lunch | GridPP PMB member. Also holding the role of ATLAS coordinator. | 20/11/2009 | LSE | The researcher discussed interesting themes that emerged during data collection in terms of middleware development, clusters of competence, power users, GridPP's relationship with EGEE, etc. This discussion was an opportunity to finalise some concepts and ideas. |
| Interview | GridPP project leader – also a member of ATLAS. | 17/03/2011 | Glasgow | The researcher discussed interesting themes that emerged during data collection. The concept of collaboration, shared visions, how this effort was coordinated, etc. were extensively discussed. This discussion was an opportunity to finalise some concepts and ideas. |
| Interview | GridPP PMB member. Used to be GridPP project leader – now involved in ATLAS dissemination effort. | 17/03/2011 | Glasgow | The researcher discussed interesting themes that emerged during data collection specifically in terms of sharing knowledge and documentation. |
| Interview | LHCb technical coordinator. | 17/03/2011 | Glasgow | The researcher discussed interesting themes that emerged during data collection specifically in terms of usability of the Grid, competition and collaboration between experiments and competing middleware solutions. |

| | | | | |
|---|---|---|---|---|
| Interview | Postdoc on ATLAS. | 17/03/2011 | Glasgow | |
| Interview | PhD student on ATLAS. | 17/03/2011 | Glasgow | |
| Interview | CMS representative to GridPP PMB. Member of the GridPP user board. Also CMS UK Tier-1 coordinator. | 25/03/2011 | Bristol | |
| Interview | Postdoc on CMS. | 25/03/2011 | Bristol | |

# Appendix C – Interview protocol

This Appendix provides a detailed list of the interview questions asked throughout the interviewing process. The interviewing process (which was thoroughly discussed in the methodology chapter) was an intense and recursive process. Not all questions were asked in every interview. The interviewing usually began with a number of general questions and then proceeded to more specific questions depending on the different roles people held in the projects. For example, there are a number of questions which only experimental end-users (e.g. question 17) could address. Furthermore, the pilot study undertaken together with the first round of interviews formed the initial findings for the elaboration of more detailed interview questions (question 18 and onwards), which led to a more in-depth exploration during the second phase of interviewing.

1) Personal role (professional background, why did you get involved in this etc.)

2) What does your job/role involve? What are the biggest challenges in your job?

3) What are your daily activities?

4) **How do you develop Grid middleware and/or applications to run on top of the middleware?**
   - ➔ What methods/practices do you use? Are you using any specific methodologies (e.g. agile methods, traditional methods?)
   - ➔ Do you design any diagrams? How do you design them and how detailed are they? UML diagrams?
   - ➔ What kind of programming languages do you employ?
   - ➔ Do you have a set of priorities, a clear statement of goals? Do you follow them religiously?
   - ➔ Do you write documentation? Who writes documentation? What kind of documents? How detailed are they? Are users allowed to make annotations/correction comments on the documents?
   - ➔ How consistent is your documentation with what you actually do in practice?
   - ➔ What kind of testing do you perform?
   - ➔ Does the way the Grid is being developed relate to your physics' tradition, culture and history?

- ➔ Could you please provide an overall picture of the whole development cycle of the Grid middleware? How does it start and how does it end? Where does the development team, integration team, ETICS team, PPS team, deployment team fit in the process? What is the hierarchy and what are the dependencies? Do you feel that this development process is quick, effective and efficient? Can this process be bypassed by certain people and why?
- ➔ How do you cope with unpredictable changes and requirements?

5) What are the advantages/disadvantages of developing the Grid middleware in this particular way?

6) How do you deploy Grid components? Do you follow a specific process?

7) What about maintenance? How would you define maintenance in this context?

8) **How do you collaborate in order to develop the Grid middleware?**
- ➔ Who do you work with? How do you communicate with them?
- ➔ What kinds of meetings do you attend?
- ➔ How do you know what work is required by you or what needs to be done?
- ➔ How do you plan your actions, what to do next?

9) How do you identify problems with your software? How do you solve them?

10) **Who are the stakeholders/users of your technology and how do you collaborate with them in order to find out what they need?**
- ➔ How do you incorporate their requirements to the Grid?
- ➔ How are users supported if something goes wrong? Is there a feedback loop with the users?
- ➔ Do you believe users play an important role in shaping the technology and the way you develop it?

11) **We know that there are middleware developers at CERN, RAL , ITALY etc. and there are different projects involved in this middleware development, e.g. EGEE, LCG, GRIDPP etc.**
- ➔ How is this large-scale distributed collaboration of developers being managed and coordinated?

- ➔ What are the TCG, EMT meetings and what kind of people are involved in them? How useful are these meetings and how quickly is their response to the problems/priorities identified in these meetings?
- ➔ How do you share knowledge, expertise, shared goals etc. within the distributed development teams?
- ➔ How and where is this knowledge codified/stored?
- ➔ How do you make sure that the different interests/motivations of the different people and projects involved are aligned in order for their actions to lead towards the shared objective which is the development of the Grid for the LHC? (maybe funding plays an important role in that?)
- ➔ How did you collaborate in the beginning in order to set-up the main elements and rules for the initial development? What decisions were taken on those early stages which might influence today's development?
- ➔ I suspect that conflicts and tensions are quite common among highly intelligent and opinionated people. How do problems are negotiated and solved and how do you reach agreements?
- ➔ How do you collaborate in order to achieve consistency and coherence when integrating the different components each development team delivers?
- ➔ How do you cope with this distributed way of working?
- ➔ What do you think are the demands/requirements of this distributed development environment?
- ➔ How the way you develop the Grid middleware responds to these demands?
- ➔ Do you believe that collaboration plays an important/crucial role in the successful development of technologies being developed in a distributed way? Why?
- ➔ Do you feel that building community bonds is crucial for making such distributed efforts work?
- ➔ If you could do it all over again, would you have done anything different?

12) We all know that the Grid by itself is a large-scale distributed system. Do you believe that this distributed nature of the Grid influences the way you develop it (methods/practices/coordination mechanisms). If you were developing a system of much smaller scale would have done it in the same way?

13) What role do you think funding plays in the development of the Grid in particular and in the management of the distributed effort in general?

14) What do you think are the differences between developing systems in industry with the way PP develop their Grid?

15) Do you see any differences with the open source community in terms of how you develop systems?

16) What do you think other e-Science projects or other disciplines could learn from the PP community in terms of the way you develop the Grid?

17) **From a user's point of view, what do you think of the usability of the Grid?**

- ➔ What are the main problems with the Grid? What challenges does an ordinary user face when using the Grid?
- ➔ What do you think about the middleware? What difficulties do you have with it?
- ➔ How do you know how to use the technology?
- ➔ Is the technology well-documented?
- ➔ What are you doing when you have problems, or when the system breaks down?
- ➔ How do you make your demands known?
- ➔ How does the Grid get embedded in your daily work practices?

### *More detailed interview questions*

18) We have been told that when you hire particle physicists you know what to expect from them because the PP's culture/tradition (e.g. self-motivation) is sort of been ingrained in them, as opposed to CS where you do not know what to expect from them. How is your culture/tradition being ingrained into new members?

19) We have been told that when PP are faced with a problem they will develop a solution that might not be well-written but it will work, while CS will do a thorough design but the end-product will never quite emerge for people to use. But in middleware development both PP and CS are involved. Therefore how do these different approaches somehow get aligned?

20) We have been told that the problem with the middleware is that it is not developed by CS who know how to properly develop software; rather it is developed by physicists who write poor quality code, they do not document their code so it is difficult for others to reuse it etc. What is your view in this?

21) Do you believe that the involvement of CS helps or hinders the development? Please explain why.

22) We have been told that PP are pragmatic in the way they work. How is such pragmatism reflected in the way the Grid is developed?

23) We have been told that there is a level of autonomy in the way you work and flexibility in changing roles (e.g. from developer to integrator, from user to developer etc.). There is also a great feeling of self-motivation and voluntarism in doing things that you are not responsible for. How do you think these characteristics help or hinder the distributed development effort?

24) We have been told that competing solutions is a very common phenomenon in the PP environment. Do you agree with this? If yes, do you think that the Grid is developed through different competing solutions?

25) We have been told that "natural selection" is the best way to ensure that the most relevant solution is carried forward. However, it is also argued that competing solutions lead to duplication of solutions and waste of time and resources. Could you comment on this?

26) Do you think that this "competing solutions way of working" reflects the need of PP to be in control and therefore preferring to build a solution from scratch rather than take something that although more general already exists?

27) Some have described that the middleware is developed through a network of patches of expertise (clusters of competence). Is this really how the Grid middleware is developed? Could you explain this further?

28) PP have been described to have a distinctive, exceptional collaboration.

- ➔ What is so unique about PP in terms of the way they collaborate?
- ➔ How such tradition is reflected in the way the Grid is developed?
- ➔ Do you believe that the way they collaborate might influence the way the Grid is developed?

29) We have been told that PP have a tradition of distributed collaborations for building detectors, but not for computing projects like this one. How do you think this affect (helps/hinders) the way you work?

30) Some have described PP's collaboration as distinctive and exceptional; however others have described it as symbiotic, or even as an ecology where there is a certain element of competition and people are seen as predators.
   - ➔ How would you describe the collaboration?
   - ➔ Do you feel that there is an amount of competition between different development teams, different countries and different projects?
   - ➔ How do you manage to balance competition with collaboration?

## Appendix D – Examples of codes and quotations and code families

This Appendix provides examples of codes constructed to help coding and analysing the data, labelling aspects of the project, the interviewees' practices, and emerging ideas. These codes are linked to relevant quotations providing a flavour of the data analysis process. Further examples of code families are also provided. Conceptual constructs related to relevant literature, such as GSD, contradictions, history/culture etc. were used to set up code families, which grouped individual codes thereby creating a network view of the data.

| Quotations | Interviewee | Related codes |
|---|---|---|
| But I think historically particle physics has this background in teamwork and this way of working.. And we've all seen this happen…when I started off in experiments, there were about 20 people in an experiment, and now there's 500 in the current experiment, ATLAS probably 2,000…[]...So it's sort of second nature. I think that's probably the advantage we have. It's more second nature to us than to other disciplines. | Chair of GridPP user board also holding the role of the UK computing coordinator for the LHCb experiment | [Collaboration], [PP tradition] |
| It is a collaboration and we have to work with people and convince them that they need to do what we are asking them to do, and they need to do it in a reasonable time. So in physics the collaborative aspect has been there for a long time, and people know how to work together, the physicists know how to work together. | LCG senior member – Particle physicist | [Collaboration] |
| It's just that we're amazing! You see it when you come to these meetings – everyone's a pretty smart person, the community is really good. You have all these smart people working towards the same goal and, at the same time, everyone is really friendly and quite willing to allow you to ask some questions to find out information. So I think there is quite a positive vibe within the community that kind of spurs people on to do a good job. Everyone's really approachable and willing to help everyone else. | Particle physicist – Storage expert | [Community], [Shared vision] |

| | | |
|---|---|---|
| I would say we, we use methodologies but just up to the limit that its appropriate for what we're trying to do…how could we possibly do a formal software engineering approach if we're going to change it. It's used up to a point but almost certainly not completely religiously. | GridPP management board member | [System development methodologies] |
| The Grid will work and the reason is that in this community there are a lot of people who are smart enough to know how to get it to work. So if there is a problem that comes up the people will stay up at night, or weekends, and figure out how to get it to work, and it may be a horrible, horrible hack, but it will work.<br><br>I think if problems did come up then there are people out there with a lot of motivation and a lot of knowledge and a lot of skill who will come up with the solutions. | LCG – Particle physicist developer | [Confidence that it will work]<br><br>[Self-motivation], [PP culture] |
| If there is some of the official grid technology which isn't working then we would just bypass that and replace it with a home-grown replacement. So our primary purpose is to analyse the data, if we can do it on the Grid, that's fine, but if it gets in the way I am sure we'll just chop it out. | Particle physicist user – CMS experiment | [Powerful PP users] |
| I think we have somehow learned how to organise things at project management level and how to take the pragmatic view and when faced with a problem to know how to get from there to the solution.<br><br>Particle physicists are very dirty programmers, they are not computer scientists and they really will use the fastest way to get at something. They do not use structured programme design unless they are forced to and they usually want the fast hack. This is what I mean with pragmatism. | UK computing coordinator for LHCb – Also chair of the GridPP User board<br><br>Particle physicist – Storage development | [Pragmatism] |

| | | |
|---|---|---|
| Particle physicists have been at the forefront of computing in the past. What they did, that was a frontier in electronics and in computing certainly – so they think that they can do any kind of thing. They think they can do anything and developing software is one of these. | EGEE computer scientist | [Arrogance] |
| So, we have slightly different goals, we all have our own physics analysis channel we want to do, but to achieve that goal, we need the detector to work, we need the machine to work, we need the collaboration to work, we need lots of input from our collaborators. So, many people at the lower levels, you know, all the graduate students, they share, embrace a hundred percent this global goal of the organization. The means of achieving that is collaboration; that's why collaboration becomes then the natural tool, much more so in our sort of type of community than, in, say, the corporate structure whereas I think the shared goal peters out after you get down to the first few layers of management. And I think that we have this history and we have learnt that collaboration works. | GridPP project leader | [Balance of competition and collaboration] |

| **Examples of code families** | **Related codes** |
|---|---|
| Recipe for successful GSD | [Advantages of distribution], [Balance of competition and collaboration], [Balance of different interests], [Clusters of competence], [Collaboration], [Co-located development], [Community building], [Identity], [Lessons learned], [Mixing structure & collaboration], [Need for both agility and discipline], [Need for more structure], [Need for stronger management], [Negotiation and compromises], [Parallel solutions], [PP culture], [Self-motivation], [Voluntarism], [What others can learn from PP] |
| History/Culture | [Community], [Democratic meritocracy], [Dependence on good people], [Developers-users], [Disadvantages of the way PP work], [Division of labour], [Evolution & revolution], [Goal of physics], [Identity], [PP culture], [PP & Computing], [PP and other fields], [PP practices], [PP tradition], [PP tradition of distributed collaboration], [PP VS CS], [PP VS Industry], [Practices |

| | |
|---|---|
| | unstructured], [Pragmatism], [Pride], [Problem solving], [Relative value of technical and managerial experience], [Shared vision], [Trust] |
| Systems development | [Advantages of CS developing PP's software], [Advantages of distribution], [Agility], [Benefits of object orientation], [Certification], [Complexity and Speed], [Deployment], [Developer-users], [Development process], [Different types of developers], [Disadvantages of distribution], [Disadvantages of the way PP work], [Documentation], [Efficiency], [Fast development], [Flexibility], [Informal communication], [Interdependency], [Interoperability], [Lack of experience], [Lack of structure], [Maintenance], [Middleware], [Nature of the Grid influences development], [Need-driven], [Need for both agility and discipline], [Need for more structure], [Need for stronger management], [Negotiation and compromises], [Open source], [Physics intuition in the code], [Programming languages], [Prototype], [Quality of codes], [Release cycle], [Scale], [Security], [Service], [Standardization of the development process], [Standards], [Sys Admin], [System Development Methodologies], [Technical management], [Testing], [Tools], [User requirements], [User Support] |
| Contradictions | [Balancing experimentation with discipline], [By-passing the EGEE formal process], [Clusters of competence], [Conflicting goals], [Division of labour], [From development to production], [Grid VS other computing], [GridPP VS EGEE], [How the Grid was developed pre-EGEE], [Ineffectiveness of EGEE], [IT evolution at CERN], [Lack of experience], [Lack of structure], [Need for both agility and discipline], [Parallel solutions], [Physics intuition in the code], [Powerful PP users], [PP's focus on their own problems], [PP's symbiotic collaboration], [PP & computing], [PP tradition influencing the project], [PP VS CS], [Pressure of the LHC demands], [Pride], [Problems with GGUS], [Problems with traditional methodologies], [Project structure], [Reactive and proactive mode], [Scalability of improvisation/pragmatism], [Sites – black and white listing], [Tensions], [Tension between EGEE & LCG] |