universität wien

DISSERTATION

Reference Architecture for Virtual Organization

Verfasserin

Wajeeha Khalil

angestrebter akademischer Grad

Doktorin der Sozial- und Wirtschaftswissenschaften
(Dr.rer.soc.oec.)

Wien, 2012

Studienkennzahl lt. Studienblatt:	A 084 881
Dissertationsgebiet lt. Studienblatt:	Informatik
Betreuerin / Betreuer:	Univ.-Prof. Dipl.-Ing. Dr. Erich Schikuta

To my Parents...

Abstract

"United we stand, divided we fall" is a well known saying. We are living in the era of virtual collaborations. Resources are logical and solutions are virtual. Advancement on conceptual and technological level has enhanced the way people communicate. Everything-as-a-Service once a dream, now becoming a reality.

Problem nature has also been changed over the time. Today, e-Collaborations are applied to all the domains possible. Medical, engineering, meteorology, biology, chemistry, physics, earthquake and weather forecast, social networks and so on, all are using electronic platforms. Extensive data and computing resources are in need and assistance from human experts is also becoming essential. This puts a great responsibility on Information Technology (IT) researchers and developers to provide generic platforms where user can easily communicate and solve their problems. To realize this concept, distributed computing has offered many paradigms, e.g. cluster, grid, cloud computing. Virtual Organization (VO) is a logical orchestration of globally dispersed resources to achieve common goals.

Existing paradigms and technology is used to form Virtual Organization, but lack of standards remained a critical issue for last two decades. Our research endeavor focuses on developing a standard for Virtual Organization building process. The proposed standardization process is a two phase activity. First phase provides requirement analysis and the second phase presents a Reference Architecture for Virtual Organization (RAVO). This form of standardization is chosen to accommodate both technological and paradigm shift. We categorize our efforts in two parts. First part consists of a pattern to identify the requirements and components of a Virtual Organization [1]. Second part details a generic framework based on the concept of Everything-as-a-Service. Stakeholders are an important entity in any collaborative environment [2] [3]. We developed a pattern for stakeholders and presented new relationship between user and resources in form of *Subject* [1] [4].

Finally, these concepts are materialized as a concrete framework in the domain of E-learning and Computational Intelligence. Stakeholders and *Subject* relationship are also verified in the domain of informal Virtual Organizations (e.g. Social Networks) [5]. For evaluation purpose an instance based on RAVO, named N2SKY [6] is developed by a master student at the University of Vienna.

Zusammenfassung

"United we stand, divided we fall" ist eine bekannte Englische Redewendung. Wir leben in der Zeit der virtuellen Zusammenarbeit. Quellen sind logisch und Lösungen virtuell. Fortschritte auf der konzeptionellen und technologischen Ebene verbessern die Weise der menschlichen Kommunikation. Everything-as-a-Service war einmal nur ein Traum. Heute wird es Realität.

Auch die Art der zu bewältigenden Probleme hat sich im Laufe der Zeit verändert. Heutzutage wird die Online-Kollaboration über das Internet (e-Collaboration) in allen möglichen wissenschaftlichen Gebieten angewendet. Medizin, Technik, Meteorologie, Biologie, Chemie, Physik, Erdbeben und Wettervorhersage, Soziale Netzwerke usw., alle benutzen elektronische Plattformen. Umfangreiche Daten und Rechenressourcen sind nötig und auch die Assistenz durch menschlichee Experten wird immer bedeutsamer. Diese Situation stellt eine grosse Verantwortung für IT Forscher und Entwickler dar, generische Plattformen zu schaffen, auf denen Benutzer einfach kommunizieren und Probleme gemeinsam lösen können. Das Verteiltes Rechnen (Distributed Computing) bietet viele technische Paradigmen an, wie zum Beispiel Cluster Computing, Grid Computing, Cloud Computing, um dieses Konzept umzusetzen. Konzeptuell erlauben Virtuelle Organisationen (Virtual Organization) ein harmonisches Zusammenspiel von global verbreiteten Ressourcen, um gemeinsam Ziele zu erreichen.

Bestehende Paradigmen und Technologie werden heute in der Praxis zum Aufbau von Virtuellen Organisationen verwendet. Der Mangel an existierenden und anerkannten Standards dazu stellt jedoch ein kritischer Punkt für die letzten zwei Dekaden dar. Unsere Forschungsbemühung konzentriert sich daher auf die Entwicklung eines Standards zum Entwurf und zur Realisierung Virtueller Organisationen. Der vorgelegte Standardisierungsansatz besteht aus zwei Phasen. Die erste Phase führt eine Anforderungsanalyse durch und die zweite Phase stellt eine Referenzarchitektur (Reference Architecture) für Virtuelle Organisationen (RAVO) vor. Dieser Standardisierungsansatz wurde gewählt um sowohl technologische als auch paradigmatische Wechsel zu erlauben. Wir teilen unsere Bemühungen in zwei Bereiche. Zuerst präsentieren wir einen Modellierungsansatz, um die Anforderungen und Komponenten der Virtuellen Organisation [1] zu identifiziert. Danach definieren wir einen generischen Rahmen, der auf dem Everything-as-a-Service Konzept aufbaut. Stakeholders sind ein wichtiges Element in jeder kooperationsunterstützenden Umgebung [2] [3]. Daher haben wir ein neuartiges Schema für Stakeholders entwickelt, die es erlaubt Beziehung zwischen Benutzer und Ressourcen in Form von Subjekten [1] [4] abzubilden.

Zum Schluss werden diese Konzepte in Form konkreter Umsetzungen auf dem

Gebiet des E-Learning und der Computational Intelligence untersucht. Die neuen Elemente der Stakeholders und Subjekt-Beziehungen wurden weiters in informelle Virtuelle Organisationen, sogenannten Sozialen Netzwerken, verifiziert [5]. Zur Evaluation des vorgestellten Ansatzes wurde schliesslich eine praktische Umsetzung, die auf RAVO basiert, unter dem Namen N2Sky als Masterarbeit an der Universität Wien durchgeführt.

Ackowledgements

In the name of ALLAH (Subhana hu Taala), Who blessed me with all what I am able to achieve in this world. I am thankful to ALLAH (Subhana hu Taala). I am thankful for my parents and siblings who were a constant source of encouragement during my whole life and specially my course of studies in Vienna. I owe a special thanks to my advisor Prof. Erich Schikuta, who inspired me both as a human and as a supervisor. His analytical and critical skills polished my research abilities and made me achieve this target. He was always there for me during my bad health patches and never lost faith in me which requires allot of patience.

My research fellows from Austria namely Helmut Wanek, Jürgen Mangler, Peter Paul Beran, Sonja Kabicher, were a great help during my research. Jürgen Mangler's critical analysis proved helpful during my research paper and dissertation writing. I am thankful to Erwin Mann for his collaboration and support during *Evaluation* of RAVO. My research fellows from Pakistan Irfan-ul-Haq and Altaf Ahmed Huqqani were a great source of help during course work.

I am thankful to my friends Zahida Parveen and Mariam Anees for keeping me motivated during my thesis writing and stay in Vienna.

I am thankful to Higher Education Commission of Pakistan (HEC) for providing me the opportunity to learn and experience the research environment abroad by funding my Ph.D studies in Vienna.

I am thankful to my colleague M. Asfandeyar for providing me discussion time and helping me in proof reading of my dissertation.

I also cordially acknowledge my research mentor, Umar Farooq for supporting me during my studies and life.

Wajeeha Khalil

Vienna, Austria

February 2012

Remarks

I use "we" and "our" in my thesis as a gesture of respect to the research community. This convention is part of my attitude towards that community. She is used to refer the user without any gender discrimination. VO is used instead of Virtual Organization in whole document. Abbreviations commonly used are listed.

Contents

List of Figures

List of Tables

List of Abbreviations

ARCON	A Reference Model for Collaborative Networks
Amazon EC2	Amazon Elastic Compute Cloud
API	Application Programming Interface
BIRN	Biomedical Informatics Research Network
BAP	Business Architecture Project
BIDSAVER	Business Integrator Dynamic Support Agents for Virtual Enterprize
caBIG	Cancer Biomedical Informatics Grid
CNO	Collaborative Networked Organization
CI	Computational Intelligence
CIML	Computational Intelligence and Machine Learning
CS	Computational Science
CEWebS	Cooperative Environment Web Service
DP	Digital Preservation
EGEE	Enabling Grid for E-scienc in Europe
EA	Enterprize Architecture
ECOLEAD	European Collaborative networked Organisations LEADership
XaaS	Everything as a Service
EXTERNAL	Extended Enterprize Resources, Network Architectures and Learning
HPC	High Performance Computing
ICT	Information Communication Technology
IT	Information Technology
IaaS	Infrastructure as a Service
IP	Intellectual Property
LHC	Large Hadron Collider
LEAD	Linked Environments for Atmospheric Discovery
NIST	National Institute of Science and Technology
NN	Neural Network
NSF	National Science Foundation
OGF	Open Grid Forum
OSG	Open Science Grid
OASIS	Organization for the Advancement of Structured Information Standards
PaaS	Platform as a Service
RA	Reference Architecture
RAVO	Reference Architecture for Virtual Organization
SOA	Service Oriented Architecture
SaaS	Software as a Service
SDK	Software Development Kit

SCEC	Southern California Earthquake Center
SCOR	Supply Chain Operations Reference
SAF	System Architecture Forum
VELOCI	Virtual E-learning Organization for Computational Intelligence
VE	Virtual Environment
VM	Virtual Machine
VO	Virtual Organization
VOCI	Virtual Organization for Computational Intelligence

1 Introduction and Motivation

"Resource/Service as a utility" once a dream, is now a reality we are living with. Utility computing is not a new concept, but rather it has quite a long history. Among the earliest references is by John McCarthy[1]:

"If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility... The computer utility could become the basis of a new and important industry."

Last two decades of Information Technology (IT) development has witnessed the specific efforts done to make this statement of John McCarthy a reality. Utility computing is providing basics for the current day resource utilization. Cluster, grids and now cloud computing have made this vision a reality. E-Collaborations also called *virtual organizations* have been evolved with the technological and paradigm shift. Cluster computing offered more centralized resource pool, while grid computing remaind in need via hardware and computation cycles offerings to the scientific community. Grid computing models observed a deadlock after the introduction of cloud computing concepts. Based on Pay-as-you-use criteria, cloud computing is still in early stage. Research efforts are going on to establish the basis of cloud computing as Every-thing-as-a-Service paradigm.

Infrastructure providing resources as a utility must be dynamic, scalable and reliable. Orchestration of resources across the globe, named as Virtual Organization (VO)/Virtual Enterprize (VE) has been extensively deployed to achieve this target. Change in the hardware and software technology, computing paradigms algorithm and procedures, incorporation of knowledge rather information and data, made the concepts of VO vague. Though VO had been created utilizing the best technology known to that time, but the success was short lived. There are three main issues, which has to be considered in order to understand:

- Advancement in hardware/software technology.
- Birth of new computing paradigms.
- Changed nature of resources and requirements from end user.

We are living in the age of transformation. A paradigm shift is one that effects the society as a whole. According to Peter Drucker such transformation place over fifty to sixty-year periods [9]. In his book "Post-Capitalist Society", he outlines three earlier periods of dramatic changes in the Western World.

[1] John McCarthy, speaking at the MIT Centennial in 1961, "Architects of the Information Society, Thirty-Five Years of the Laboratory for Computer Science at MIT", edited by Hal Abelson

- The rise of medieval craft guilds and urban centuries. Long distance trade (thirteenth-Century Europe) [9].
- The renaissance period of Gutenburg's printing press and Lutheran Reformation (1455-1519) [9].
- The industrial revolution, starting with Watt's steam engine (1776-1815) [9].

Drucker describes the current shift, which he reckons started around 1960 and will continue around 2010 or 2020, as follows [9]

..."We are entering the knowledge society in which the basic economic resources no longer capital, or natural resources, or labor, but is and will be knowledge and where knowledge workers play a central role"...

Existing technologies and paradigms do not vanish with the birth of new concepts rather they adopt what is positive and remove what is not required. Technology and paradigm used to form VO have also faced this transformation.For example networking, distributed computing, cluster computing, grid computing, utility computing and now cloud computing, all are related and are improvements of the existing concepts. When technology changes or improves, paradigm needs an upgrade too. New methods and algorithms are created to support the hardware. Another main factor is the requirements from the user community. The user community puts a demand on the technology and computing paradigm and they evolve accordingly.

"Resources/Services as a utility" is main theme of collaboration. To achieve the goal(s), organizations and individuals gather all the resources available. The spectrum of availability has covered the whole globe. Today, time and space are not a limit due to Information and Communication Technology (ICT) advancements. This revolution has an impact on the resources types. Initial collaborations offered only storage and downloading (Sethi@home[2]: P2P networks), computing cycles and storage space (grid computing and cluster computing). Main focus remained at hardware and software sharing, but VOs for scientific research initiated another requirement, i.e. need of a human expert to guide the beginners in the said domain. Expert becomes an integral part of collaborations. Also, the two way contribution (duplex) motivated us to review and categorize the resource in the vicinity of VO. The categorization we presented is also vigilant to depict the general pattern of resources in any domain.

VO is the right place for both technology and computing paradigm to merge and achieve the objectives. In the past two decades collaborative computing has remained main concern of technology produced. Optimization of time and heterogeneous resources by building VO is the key point of today's research directions. Vision of a VO has evolved with the networking and distributed computing concepts.

Research community recognizes VO with different names, e.g. collaboratories [10] [11], E-Science or E-Research [10] [12], distributed work groups or virtual teams [10] [13], virtual environments, virtual enterprize [7] and online communities [10] [14].

[2] http://setiathome.berkeley.edu/

Initially, focus was to improve business by utilizing the ability to gather resources which are scattered across the dimensions of time, space and structure. With the advent of modern technology, VO has encompassed almost all fields of life. We can say that every human will be soon part of a VO. VO's concepts need to be revisited with this evolution in general. VOs have been visioned from the business perspective in early 1990s. Pervasiveness of technology and improvements in computing paradigms has extended the domain of VO to cover all the areas where individuals and organizations meet to achieve some goal (formal Virtual organization) or without any specific common objective, e.g. Social networks (informal VO). To the best of our knowledge, till now there are no standard procedures or patterns for how VO should be created and evolve to accommodate the changes in its integral parts or entities. Lack of standards for VO motivated us to provide a standard vision of E-collaboration incorporating both paradigm and technology shift as a Reference Architecture (RA) to achieve common objective(s) in any domain. Our research efforts also introduced new concepts regarding resources and stakeholder of a VO.

To provide a standard for VO, we consider the existing technologies and paradigms. Service Oriented Architecture (SOA), Web 2.0 and Web 3.0 are the underlying technological platform, and computing paradigms include utility computing and cloud computing.

During our research process we studied the existing infrastructures available for VO. Utilizing electronic collaborations for achieving common goals is a tradition rather a requirement. Distributed resources are gathered using an infrastructure and are exploited to obtain the said results. In IT world such collaboration is known as VO. Idea is to provide resources as a utility to the end user. Service-Oriented infrastructures need to act dynamically to fulfil the demands from organizations and businesses. We encountered the following addressable issues:

- Does existing electronic collaboration approaches follow a standard?
- Can we define patterns without predefined standards?
- Does existing infrastructures fulfil the requirements of participating entities?
- Are the existing infrastructures dynamic and adaptable to the rapidly updating IT and business world?
- Can we design a generic platform to integrate resources from multiple domains using essential and optional parts?

Our research aims to answer these questions. VO's creation process lacks standards/patterns/methods [10]. We analyzed existing VOs and the process of their creation through available documentation. We found the following answers to the above questions:

- Currently, there exist no specific standards for building VO or E-collaboration.

- Existing infrastructures are modified for specific domain needs and cannot be generalized to all the domains. Since, existing technology is used without following any standard for creating a VO, it is hard to foresee incoming demands from the participants.
- We require a generic platform to integrate resources from a single domain or multiple domains.
- Defining a generic platform on the basis of Everything-as-a-Service (XaaS)[15] concept is a solution. Definition of participating components as:
  - Essential parts
  - Optional parts

The last answer laid basis for our research. Defining standards for VO is a solution to the questions, raised above. Obstacles to reach our goal were:

- Lack of documentation of building process of a VO.
- Definition of stakeholder and their roles.
- Overlapping resource categorization.

These Obstacles laid foundation for our research work. This thesis is an answer with examples of solution. We present the following solutions for these obstacles:

- Generalized patterns for building a VO.
- Defining components of a VO.
- Providing new definitions and examples of *Resources* and *Stakeholder* in different domains and justifying them in real world.
- Presenting a Reference Architecture for Virtual Organization (RAVO) which can be applied as a starting point for any community (belonging to a single or multiple domains) to collaborate.

1.1 Use Case

The use case is detailed in the context of cloud SPI Model (Software-as-a-Service, Platform-as-a-Service, Infrastructure-as-a-Service, e.g. SPI) [16], supporting XaaS platform for a VO. Existing approaches used for the creation of VO are domain specific and are limited by lack of standards. Therefore, focus areas of this use case are:

- Creation of a generic platform based on SPI Model.
- Identifying components of a VO.
- Categorizing the services in SPI Layers.
- Identification and definition of Stakeholder in VO.

1.2 Scenario

RAVO is envisioned to provide a guideline for creating a VO in any domain. It promises to support evolution of existing systems or building VO from scratch. We present scenarios based on the use case described in section 1.1.

1.2.1 Building Virtual Organization for Computational Intelligence

A student needs to predict market directions using N2Grid system (currently evolving to N2SKY [6] under development) developed at University of Vienna. Current offerings of the system are:

- Graphical interface.
- Selection of paradigms.
- Computation resources.
- Free of cost resource utilization.

User computes the problem with her own data but results are not satisfactory. She requires an expert's opinion and a wider range of data sources. Thus she needs a portal containing specific resources and expert's help to solve this activity.

Solution: Upgrade N2Grid System as a Portal by integrating required components. Results of up-gradation are:

- Solve the specific problem.
- Scale the system to have a VO for Computational Intelligence (CI).
- Find the way to standardize the efforts to build a VO.
- N2Sky offers a business platform by integrating a business model. Providing a profitable collaboration environment for VO.

1.2.2 Identification of Stakeholder According to Current Needs in E-learning

Current E-learning systems focus more on what an E-learning environment provide to the user. These systems usually target undergraduate and graduate level students. How a research student can be presented in an E-learning environment?

Solution to this requires:

- Understanding and redefining the roles of stakeholder in E-learning systems.
- Focusing on requirements and activities performed by research students.
- Deploying research *students* and *teachers* as a resource in the E-learning environment rather a consumer.

This can be achieved by deploying the E-learning resources as a VO, where students and teachers are considered integral part of the environment.

1.3 Outcomes

These scenarios give rise to the need of building a generic standard for virtual collaboration of resources. The standard for VO which:

- Identifies its components.
- Presents stakeholder's clearly, incorporates new user roles in the alliance.
- Supports formation and management of a VO either on temporary or permanent basis.
- Include a business perspective for the business community.

1.4 Goals

This thesis presents a Reference Architecture for Virtual Organization (RAVO). This reference architecture is not restricted to a specific type of VO or to a specific problem domain. It is a generic architecture build on the basis of XaaS and SPI model. This reference architecture also supports integration of resources from multiple domains.

We have introduced a unique term *Subject* for the stakeholder in the overlapping area, where a stakeholder itself is available as a resource to other stakeholders [1]. VO is redefined in terms of *Subject*. Resources are categorized as logical and physical to include the human expertise as a resource in the VO. During our research endeavor, existing VOs were studied and analyzed. We also introduced a pilot approach to evolve the existing system resources into a VO [1]. We have chosen *Computational Intelligence*, *E-learning*, *Social Networks* and *Computational Science* as target domains for analyzing and testing our research findings.

1.5 Organization of the Thesis

In chapter 1 we detailed motivation behind our research work, main issues and concerns are presented in Question/Answer format.

Chapter 2 presents an overview of the state of the art of Virtual Organization (VO) and Reference Architecture (RA). It briefs the basic concepts and related work.

Chapter 3 details the process of building a Reference Architecture for Virtual Organization RAVO). First part, *requirement analysis phase* provides a justification to different critical questions. It also identifies the main components of a VO. It details the process of generating a general pattern for recording requirements thereby providing a basic pattern for building VO.

Second part explains the architecture building process. An overview of the existing architecture is also phrased. Proposed framework, stakeholders and Viewpoints are detailed. Interface specification for components at different layers is elaborated. Mandatory and optional components are identified

Chapter 4 explains the relationship among stakeholder and resource in the context of VO. It also presents the stackholder and resources categorization. Application of RAVO to informal VO is also detailed.

Chapter 5 describes the application domains and brief introduction of the supporting system in said domains. Example use cases of RAVO are demonstrated in the domain of Computational Intelligence, E-learning and Computational Science.

Chapter 6 explains the evaluation of the RAVO. Quantitative analysis by a senior researcher and a master student at University of Vienna developing a cloud based, Neural Network Virtual Organization named N2SYK applying the RAVO framework is presented.

Chapter 7 concludes our research efforts.

Appendix A lists the analysis how RAVO supported the development of N2SKY in different phases and provides an elaborated comparison. Appendix B presents the Research Statements and Appendix C explains research publications and their contribution to the dissertation.

2 State of the Art

This chapter details the foundation blocks of our research focus. First part explains the basic concepts of Virtual Organization (VO) and second part elaborates the Reference Architecture (RA). Technological issues and related computing paradigms are also briefly explained.

2.1 Virtual Organization

2.1.1 Definition

VOs are rapidly growing phenomenon catalyzed by IT advancements. VO existed in different forms in last two decades but still we do not have a standard unified definition. Main reason for this lack is that every organization or group has used this concept of collaboration in the manner it was easy for them in the range of available resources (IT and human). Typically three aspects seem to change from one definition to another. First is the issue of whether a social dependency or computer-mediated arrangement is central to the definition. Secondly, the time frame is an issue: some definitions hold on to the temporary nature of VOs while others see VOs as more permanent arrangements. Thirdly, depending on whether Virtual Enterprise/Corporation (VE) or VO is used, the definition is aimed more at either profit-making and business or inclusion of non-profit institutions respectively.

VO is a nonphysical, communication model whose purpose is to achieve a common goal. It consists typically of a heterogeneous collection of people and organizations with respect to geographical limits and nature. The term *Virtual Organization* specifies a detailed non-physical problems solving environment. Many definitions have been presented and various terms arose, e.g. collaboratories [10] [11], e-Science or e-Research [10] [12], distributed workgroups or virtual teams [10] [13], Virtual Environments, virtual enterprize [8] and online communities [10] [14].

Initially, VOs were considered to be useful for business industry. Focus remained on how to change the hierarchical structure of the organization to decentralized manner to achieve more benefits. Earlier definitions of VO focus more on business and marketing. We detail some definitions from the 1990s here by different researchers and industry.

For Byrne, "a VO is a temporary co-operation of independent companies, suppliers, customers, even erstwhile rivals, linked by information technology to share skills, costs and access to one another markets"[17] [18].

For Weber and Walsh, "the purpose of VO is the optimal use of opportunities which derive from the market and/or from resources"[19].

Dr. Bernhard R. Katzy define VO in recursive manner by saying " VOs are frequently restructured, sustained to capture the value of market opportunity and dissolved again to give way for the creation of a next VO from with in the network of independent partners" [20].

For B. Travica, "VO refers to a new organizational form characterized by temporary or permanent collection of geographically dispersed individuals, groups or organization departments not belonging to the same organization - or entire organizations, that are dependent on electronic communication for carrying out their production process "[21].

According to most agreed upon the definition, presented in year 2008, "VO is sharing the geographically dispersed resources for achieving a common goal. A VO can comprise a group of individuals whose members and resources may be dispersed geographically and institutionally, yet who function as a coherent unit through the use of cyber-infrastructure (CI)" [10].

Based on theories in management and information systems, organizational science, backed with empirical cases of VOs, researchers have presented various characteristics. Building blocks, dimensions and types of VO are hard to distinguish in existing literature. VO spans with different characteristics in multiple domains. We combine concepts from the existing information and produce a analytical categorization of the dimensions/building blocks, while types are detailed in a separate section 2.1.4 [22].

2.1.2 Building Blocks

Understanding the characteristics and distinguishing dimensions of VOs provides guidance in the design of VOs. VOs, regardless of their types and operating mode, life span, possess some specific traits. They are distributed across [10]:

- Space with participants spanning locales and institutions: Virtual collaboration creates the opportunity to disperse organizational activities. Information and communications technology (ICT), especially the Internet, makes it economically viable to separate operations and people that were previously together. Conversely it also allows organizations to aggregate operations that were previously dispersed, such as customer service through back-office call centers [23]. Other ways of reconfiguring organizations through the dimension of space include dispersed teams, and individuals who telework from a remote location.

- Time with asynchronous and synchronous interactions: In the time dimension organizations can shift operations according to the time zones. For example, engineering companies pass work in progress from one location to another around the world to do 24 hours a day design. Another use of time is the flexibility of time used by teleworkers to mesh their business duties with their lifestyle and domestic needs [23].

- Dynamic structure and processes at every stage of their life cycle, from initiation to termination.

- Computationally enabled, via collaboration support systems including email, teleconferencing, telepresence, awareness, social computing, and group information management tools.
- Computationally enhanced with simulations, databases, and analytic services that interact with human participants and are integral to the operation of the organization.

Space, time and structure are detailed as primary dimensions in literature [23] [22] [24] [25]. As opposed to the VO, time and space dimensions are constrained in traditional or "real" organizations. Time constraints occur in real organizations due to the operational time dimension of such organizations, while space dimension occurs due to constraints of location. The above described characteristics are common to all types of VOs but there are many other aspects which have been added to the list by researchers. Data, information and now knowledge has become an essential part of computing. Today's research efforts focus more to capture and process the knowledge in problem solving environments. According to David J. Skyrme [26] [25] (Chapter 4 and 8 of the referred book for details). VOs operate in the physical world of three dimensions but they also have dimensions on their own. Like the physical world where the extra dimensions of time and gravity distort space, similar distortions take place in the world of VO. VO has a different meaning for different people, depending upon its use the primary dimension changes accordingly. Skyrme lists *space*, *time* and *structure* as primary dimensions and adds *knowledge* and *cyberspace* as secondary ones [26]. He presents *knowledge* as the fourth dimension and listing it most critical to the current economy. VOs and teams come together partly because of location and other resources, but most commonly because of the unique knowledge that each party possesses. Yet the knowledge dimension of a collaboration is often neglected. Major concerns regarding *knowledge* in the context of VO are:

Who owns the knowledge generated within a VO?

Who can exploit it and how?

According to Skyrme [26] [25],the fifth dimension is the *cyber* dimension i.e. the Internet where location is imprecise, where time seems to run faster, where knowledge flows freely but haphazardly. He further breaks up *cyber* dimension into following three parts:

- Cyberspace: The location of much activity is location independent. Clients dealing with an organization often do not know their whereabouts. While cyberspace makes operations and marketing on a global scale much easier, it does create difficulties where it touches the real world. Thus, where is the point of a transaction for legal and taxation purposes? How can clients get redress if something goes wrong? A key benefit of cyberspace for the VO is that its size can be disguised [1]. It can appear to be a large corporation when it is not. What matters is how effectively it operates and performs using the medium.

[1] http://www.skyrme.com

- Cybertime: By exploiting technology and global locations, VOs can automate many of their activities for 24 hours-a-day fast turnaround operation. It means that they can collapse time by parallel processing, e.g. using shared documents, activities that are sequential in the real world.
- Cyberknowledge: Explicit knowledge can be made more easily accessible for clients. Virtual meetings and consultations can take place using videoconferencing. Cyberknowledge is more diffusible. Its also provides VO memory. The VO's knowledge base can be distributed throughout the Internet and using the proper safeguards can be protected and made accessible for VO participants. With cyberspace providing a marketplace for trading and disseminating knowledge, a new way of enhancing or adding to the VO's products and services.

Detailed concepts about VO are gathered in the report "Beyond Being There" [10] and the analysis carried out during VOSTER Project [24]. VOSTER project also includes business processes, business model, management roles, change in the VO and its source network as dimensions of VO extracted from the analysis of VOSTER a European perspective. Information presented in this section 2.1.3 is based on the above described sources.

2.1.3 Organizational Dimensions of Virtual Organization

ICT provides wide scale support for the VOs. Limitations and shortcomings, in mimicking the real world organizations as a VO, are decisive when it comes to represent the organizational structure, decision making and dynamic nature. This section addresses these aspects in detail.

2.1.3.1 Structural Aspects

VOs differ from traditional organization in many respects, of which structure attains main focus. VOs offer flexible structures, as compared to traditional organizations, that bring together different people and competencies to perform specific tasks. People may be in temporary teams or VOs, that exist for as long as they are needed. Sometimes these virtual teams and organizations have a degree of permanence. In other cases they may exist only for the duration of a project, or a problem to solve. VOs have a distributed architecture as opposed to the traditional ones, where hierarchical approach is applied. The shift from conventional to VO requires a basic reorientation of management philosophy. VO is open for all types of organizations because the paradigm applies at the task level, and meta-management may be elaborated centralized or decentralized way. VO can have both centralized and decentralized control structures, units and functions [27].

As described in [24], collaboration gives rise to the fundamental requirements of labor division into tasks and the coordination of these tasks. The structure of an enterprize is reflected in a way that divides its labor into distinct tasks and then achieves coordination among them. VOs literature to date (Kúrúmlúoglu et al [28]; Rezgui and Wilson, 2005 [29]; Zigurs, 2003 [30]) and research carried out within the context of E-MMEDIATE, eCOGNOS, GOBEMEN, OSMOS projects[24], has focused on the necessity of restructuring traditional organizational structures to exploit the fast development of ICTs. It emerges from the analysis of findings from the VOSTER project [31] that further research should address:

What structural work arrangements are best suited to the work that must transcend geographical boundaries and time?

How organizations effectively enforce these structures?

What are the necessary abilities of the manager to facilitate communication among team members to create clear structures and foster role clarity to improve collaboration?

Are there other strategies that organizations can implement to improve virtual team working performance?

2.1.3.2 Dynamic Decision Making and Perception

Dynamic nature of the VO has made them more complex to operate. VO should be consistent with a variety of decision making approaches in order to satisfy the requirements. Organizations find themselves in an almost constant state of change, as they strive to respond to the pressure of the increasingly globalized and competitive environment. Thus, quick decision-making and innovation activity in response to rapidly changing conditions and demands is necessary. While researchers (e.g. Barrett and Sexton, 2006 [32] ; Pawar and Sharifi, 2000 [33]) and proposed approaches PRODCHAIN, e-COGNOS, ProDAEC in this area has been unable to break away from the traditional models [24]. Rezgui and Wilson (2005) thoroughly reviewed existing barriers and argued for new approaches. Future research in this area poses the questions of [24]:

What tasks enable perception, awareness, and preparedness to change?

Do traditional managerial change mechanisms remain applicable in the VO alliance environment?

Either wise, what are the most appropriate change mechanisms?

What business and organizational methods offer innovative and sustainable services along the collaboration?

What formulas, depending on the nature and scale of the organization changes, are effective for decision making?

What is the necessary vision and systemic thinking required to manage the change life cycle?

2.1.3.3 Legal Aspects

Developers of VO typically do not have experience with governance, policy administration, and contracts. Participants of the “Beyond Being There” [10] suggested that new VOs be provided with documents that suggest,

- purpose of the collaboration, the investment of the individual partners,
- those responsible and accountable within the collaboration organizations, and

- a high level adherence policy.

Security policies for VO users should also be the part of legal document. VO consists of legally independent individual(s) and organization(s) which can leave and enter the virtual organizational boundary during a problem solving activity. The fact that a VO has a legal identity does not mean that claims cannot be addressed directly towards the members. However, claimants will probably suffer some difficulties in determining the exact identity of the different members because of the appearance of the VO as one enterprize [34]. A group of researchers in the eLEGAL[2] project implemented legal support tools and promoted an enhanced business practice in which the use of ICT in inter-enterprize information exchange is contractually stipulated. eLEGAL develops software tools for contract editing and configuration together with a virtual negotiation room [24].

Till now these legal aspects are unattended. Another noticeable issue, which needs keen attention is Intellectual Property. The use of utility based infrastructure may, to some degree, alleviate at least one area in which Intellectual Property complicates collaboration agreements. At present, many universities are so keen to encourage technology transfer revenues that they make partnerships too complicated. They disallow university employees from making software free under open-source standards such as the GNU (GNU's Not Unix) general public license [10]. Some legal aspects which need attention are deduced from VOSTER project [24] analysis are listed here.

How to manage Intellectual Property rights and cope with copyright and confidentiality issues How to manage responsibility?

How to share and distribute liability?

How to monitor these throughout collaboration? How shared responsibility by means of rights and ownership of outcomes is identified?

How these foundations can be blended together to generate the basic building block to deliver sound legal entity?

2.1.3.4 Trust

A VO is inherently dynamic in terms of goals, structure, control, resource deployment, etc. The dynamism is driven by a project structure. Tasks are accomplished in self managing, temporary project teams. The concept of "virtual" implies continuously changing interfaces and boundaries. Virtual forms are used when an organization faces unanticipated needs that must be fulfilled in short cycles. In this context there is no time to bring people physically together. Many scholars who study VOs treat them as a panacea for problems of traditional organizational forms. However, the virtual form has as many problems if not more as the traditional organization forms. Regardless of how committed and well-meaning people are initially, they tend to lose their commitment, suffer from role overload and role ambiguity over time which in turn increases free loading, absenteeism, and other negative behaviors, all of which translates to lowered project performance.

[2] http://cic.vtt.fi/projects/elegal/public.html

In a VO, trust is the heartbeat. Only trust can prevent geographical and organizational distances of team members from turning to unmanageable psychological distances. Only through trust can members be assured of other's willingness and ability to deliver on their obligations. Initially, trust is developed on the basis of transference and intentionality processes. However, as the team communicates over time, social information is gleaned from the communication exchanges and the team members will rely more and more on predictability and capability processes.

In the flourishing markets of electronic commerce, trust plays an even more important role. In order to get customers and make profit, a business should keep its system as open as possible, and make its services easy to use. However, too much openness can quickly become a security nightmare. Decisions to trust, despite the risks are already being made simple to cope with the situation. Without a framework for organized trust management, the decisions tend to end up being made "on the fly" by people whose interests lie elsewhere and who may never see the long-term consequences of those decisions. An article in "Harvard Business Review" more than a decade ago suggested that virtual teams cannot build trust [10]. While more recent research suggests otherwise, building trust within a VO certainly takes a long time and, because it is dynamically reevaluated with each interaction, remains fragile [35].

Trust is built on a foundation of interdependence and interaction that reinforces a sense of shared identity and familiarity [36] [37]. For example, when people see others executing their roles competently, predictably, and reliably, that builds trust. Trust in VOs may be different than trust in physical organizations, however, and therefore presents opportunities for considering how trust can be built other than through familiarity. Finholt and Birnholtz [38] [10] have shown that differences in professional cultures increased the chances for misunderstanding and mistrust. Overcoming genuine distrust in virtual teams, however, is a subject that remains to be explored further [10].

VOSTER projects analyzed trust and socio-cultural dimension of VO combinely[24]. The legitimate question regarding virtual teams and trust management is

whether virtual teams can function effectively in the absence of frequent face-to-face communication?

Further research issues pointed by VOSTER projects should address the following:

What facts pave the way to foster swift trust?

How is trust maintained? What working infrastructures utilized by teams attempt to foster trust? Which, if any, team training accustoms expert team members in their fields to the particular requirements of virtual working?

What can relationship management do to foster teams of mixed experiences? How would members relate and identify themselves to their manager in a virtual context?

What are the qualities that a virtual team manager ought to have to cope with the complexity resulting from non-collocation and virtual collaboration including trust, lack of cohesion and resolving issues?

In the worst case scenario, what requirements the team needs to benefit from the *diversity and dispersion regardless of trust?*

2.1.3.5 Economical Aspects

VOs can be profitable and non-profitable. Traditional organizations have been taken over by internet based collaborations. ICT developments brought new methods and paradigms to make business more easier. Adaptation of these new measures to do business in agile and innovative way is the need of the day. As a result, VOs have the potential to improve quality and performance and leverage capabilities [31]. Literature provides evidence that VO is a suitable paradigm to support the shift from traditional to Virtual Environment (VE). Economic activity in this context means the cooperation of production ingredients to achieve competitiveness and maintain good cooperation between members of the organization alliance [39] [40]. While a number of studies [39] [41] [40] [42] [31] and research carried out within the context of the ICCI, ISTforCE, BAP, BIDSAVER, E-COLLEGE, EXTERNAL projects [24] discussed the collaborative networks' economic dimension to enable organizations to realize the value of business innovation. The complex business environment poses persistent problems to organizations [24]. From the economic standpoint, achieving competitiveness and maintaining good cooperation cannot depend solely on mutual faith. Questions raised here are

Research is needed to devise how to share profits and losses in the context of an organization alliance (in terms of cash, a resource, or skill)?

How to ensure that the collective financial gain of the organization alliance outweighs the individual profits of associated member organizations?

How organizations evaluate and determine the right economic costing in a consistent manner across the network?

In our opinion, *Legal and Economical Aspects*, *Dynamic nature* and *Trust* are interdependent. We suggest that these features can be well incorporated in a *Business Model* (for profitable) or a *Trust Model*(for non profitable) VOs based on purpose, role assigned, degree of participation of collaborating entities.

2.1.3.6 Socio-Cultural Aspects

Socio-cultural barriers and limitations of maintaining virtual working teams are highlighted by integrating present literature and results from the field work. It identifies the important socio-cultural challenges inherent to the virtual business mode including issues related to trust, social cohesion, team member structure (user/manager relationships), influences on the management and strategies [24]. It emerges from several research theories that face-to-face interaction has a direct impact on organization performance through building team trust and enabling team members to exchange valuable socio-cultural information. Researchers stress the need for initial face-to-face meeting to provide the grounds for a worthwhile ICT collaboration [41] [43]. Extending this idea even further, research suggests that virtual team members conduct periodic face-to-face meetings [43].

It is essential that team managers play a vital role in favor of relationships [44] [45] [33] [29] [46]). Relationship management ought to influence a strategy that identifies and maintains relationships which in turn ensures that objectives meet expectations [47].

It is established from the analysis of findings from the VOSTER projects that organizational culture is a critical factor to hold VOs. What remains unclear are:

How team members in a virtual context build, sustain and strengthen culture in the absence of frequent face-to-face interaction?

How often should the team members communicate to remain glued?

How to foster a culture of extensive collaboration? What behaviors inhibit a team's ability to develop a shared culture?

What behaviors raise a team's ability to develop a shared culture? What current organizational culture circumstances hinder team effectiveness in the VE?

Can a set of cultural attributes that promote effectiveness of teams be identified?

How can these attributes be effectively enforced in VOs to ensure that members remain glued?

2.1.3.7 Technological Aspect

Previously researchers have associated different computing paradigms with VO, but for us VO is an abstraction of collaboration over electronic platform. Grid technology, High Performance Computing (HPC), and SOA are associated with VO. In our opinion these are the underlying technologies which support the formation of VO. To enable the concepts of VO, technology is the tool. Both are dependent on each other. Spectrum of technology in the context of VO starts from simple email to complex multi-modal Web conferencing, from shared networks to grid and cloud computing. It utilizes the best existing technology to facilitate the participants in achieving the said goals. Technology includes, hardware, software, computing paradigms, methods and procedures and frameworks. VO is based on distributed computing. VOs need technology to function and are themselves often concerned with the development of technology. VO utilizes Internet as a platform for electronic collaborations. Introduction of enhanced Web technologies has a great impact on how people communicate today.

Advanced networks between universities and research institutions support and demonstrate state-of-the-art technology using high definition video conferencing, data sharing, data visualization, and even virtual reality immersion that comes close to "being there". On the lower bandwidth spectrum, tools for course management (Sakai[3], WebCT[4], Blackboard[5], Moodle[6]), multimodal Web conferencing (WebEx[7], Microsoft NetMeeting[8]), and instant messaging or video (Microsoft Instant Messen-

[3] http://sakaiproject.org/
[4] http://webct.nottingham.ac.uk/webct/entryPageIns.dowebct
[5] http://www.blackboard.com/
[6] http://moodle.org/
[7] http://www.webex.com/
[8] http://www.microsoft.com/

ger[9], AOL Instant Message[10], Skype[11], Jabber[12]) offer environments and tools to facilitate synchronous and asynchronous communications. Grid technologies enable the federation and remote use of diverse resources, and grids are in turn supported by "middleware" [10].

As these technologies become more stable and accessible, we see new opportunities to build and use common infrastructure, thus achieving economies of scale and reducing the cost of creating new VOs. There are two basic approaches, available for achieving this goal. First method is to plan a predetermined system that thoughtfully integrates resources with top-down notions of how they will be used. The second approach is an emergent model that assembles technology that brings people together and then creates more structures, once it is evident how they are optimally using the technology.

An exemplar of the former approach is TeraGrid[13] [48], the National Science Fund (NSF)-sponsored scientific discovery infrastructure that provides an integrated computational resource through 11 partner sites. TeraGrid connects high performance computers, data resources, analysis tools, and high end experimental facilities through high-performance network connections. Although TeraGrid has added new partner sites since its foundation in 2001 (and continues to do so), and the resources provided at these sites are heterogeneous, the system is carefully coordinated through the Grid Infrastructure Group, working in partnership with the resource providers. Depending on the nature of the resources, the systems make use of shared middleware thereby providing unique resources. In this way, TeraGrid provides consistency that can be exploited for grid computing while also allowing for users with more specialized needs [10]. Similar efforts are namely Open Science Grid (OSG) [49]. OSG is structured as a community of communities, and its functionality is driven directly by the science stakeholders. Thus, while the OSG resources provide a standard software toolkit, VOs are free to add software to support their own needs. Both TeraGrid and OSG support science gateways.

At the opposite end of the spectrum, exemplars of a more lightweight approach than grid enabled environments are popular, Internet based social networking systems, one of which is Facebook[14]. It was launched in early 2004 to connect students within recognized education institutions, and today it also includes members of recognized companies and nonacademic institutions. Though it began as a hobby project in a dorm room at Harvard, it spread quickly to Universities across the world, and claimed more than 59 million users by the end of 2007, while growing at a rate of 250,000 new registrations daily. Within this free, ad-supported system, users create personal profiles, through which they can connect with friends, post photos, write blog entries, form groups, plan events, and play with a variety of free widgets built to work with the site. These widgets are an interesting illustration of the emergent quality of Facebook. The company opened the internal workings of

[9] http://explore.live.com/messenger
[10] http://www.aim.com/
[11] http://www.skype.com/intl/en/home
[12] http://www.jabber.org/
[13] http://www.teragrid.org
[14] www.facebook.com

its Application Programming Interface (API) to developers who can now develop additional tools that members can add to their profiles. Informal VO emerged as the product of Web 2.0 with the title of social networking. These collaborations are rarely follow a specific goal, rather they have multiple goals or individual objectives. Emerging Web based technologies (MySpace, Facebook, Flickr, YouTube, and Second Life) have changed how people congregate, collaborate, and communicate. In this context VOs are more like "containers" rather than "vehicles" of collaboration in that they are not necessarily driven by common goals or comparable inputs [10].

VOSTER projects established that a technological solution in the context of VO has to support the central business processes. It must support integration os systems, interoperability between disparate applications and interaction management between individuals and teams. VOSTER identified key limitations, which hinder the full exploitation of web services as a promising middleware technology to support virtual team working [24]. These limitations are listed below:

- "Existing service description and Web Service flow languages are ill suited when addressing the dynamics and nonfunctional characteristics of distributed business processes. The current Business Process Execution Language (BPEL) version does not support run-time alterations to address unforeseen problems, such as the replacement or addition of a new Web Service. In order to manage this uncertainty, BPEL processes need to have the ability to be extended to meet unforeseen post-deployment requirements and user needs".

- "Web service flow engines, such as the ones implemented to support BPEL, lack execution monitoring functionality to manage the running process. These can help debug processes during development stage, with monitoring, and even be driven by agents at production stage. It is possible, for example, to embed, without modifying the engine implementation, a planner on the top of the latter. From events triggered by a monitor, this planner can take actions to avoid any disruption and to adjust the process. Such a tool can be useful particularly for long running processes".

- "Web service composition methodologies have a focus on syntactic integration and therefore do not support automatic composition of web services. Semantic integration is crucial for web services as it allows them to (a) represent and reason about the task that a web service performs, (b) explicitly express and reason about business relations and rules, (c) understand the meaning of exchanged messages, (d) represent and reason about preconditions that are required to use the service and the effects of having invoked the service, and (e) allow intelligent composition of web services to achieve a more complex service".

- "long running virtual team processes are subject to evolutions and change of different nature: process model evolution due to change in the environment (change in the law, change in the methodology), process instance evolution (or ad-hoc evolution) due to specific events occurring during a given process execution (delay, new available or lack of resources) or partnership evolution at execution time having an impact on part of the process".

- "new forms of software licensing are needed to provide a better software service that includes configuration, maintenance, training and access to a help-desk to ensure that SMEs are efficiently supported along their path to engage effectively in virtual teams".

No matter what approach is followed, creating an integrated infrastructure is a difficult and costly endeavor, both technically and socially. It requires enormous investment of time and effort on behalf of participating entities on cobbling together, launching, and sustaining VOs. Integrating dispersed resources and people is hindered by lack of standards, adoption and acceptance across the globe. Different types of potential needs, interests in governance arrangement, copyright issues, scalability and dynamic environment (security, membership, QoS, reliability) are the critical aspect which require attention while choosing the right technology. Another important aspect is whether an existing system is taking a new transformation or VO is created from scratch. Life span also has in impact on choice of technology. In case of existing system evolution, it is more difficult to retain the integrity and consistency while staying transparent to user community.

2.1.4 Types

Types of VO are hard to classify because there are several aspects a VO can be categorized by. VO differ from each other by purpose, mode of operation, underlying topology and life span. There exist no clear demarkation or classification of VO found in [10].

- Formal vs Informal goal oriented or objective less (e.g.LEAD[15] vs Facebook).
- Temporary vs Long term Life of VO (e.g.VOSTER[16] vs TeraGrid[17]).
- Static vs Dynamic operating mode(CIML[18] vs BIRN[19]).
- Profit vs Non-Profit business oriented (AMAZON EC2[20] vs MyExperiment[21]).

According to Bredt [50] some of the agreed upon types discussed in literature are:

- "the alliance organization that emphasizes on core competencies leveraging the strengths of the people",
- "displaced organization where people are connected through internet technology but distributed geographically",
- "invisible organization that is network of call-centers and back offices where business is executed telephonically" and

[15] http://portal.leadproject.org/gridsphere/gridsphere
[16] http://cic.vtt.fi/projects/voster/public.html
[17] http://www.twgrid.org/
[18] http://www.cimlcommunity.org/
[19] http://www.birncommunity.org/
[20] http://aws.amazon.com/ec2/
[21] http://www.myexperiment.org/

- "fourth type is a truly VO that is the blend of the other three types of organizations and the best example for this is the online Amazon.com bookstore".

2.1.5 Virtual Organization Topology

A major distinction between different VOs is their underlying organizational topology, i.e. the structure of links between the different entities. According to Kúrúmlúoglu et al [28], the structure of VO has been viewed using three different types of topologies:

- A Supply-chain topology: VO in supply chain networks, which is characterized by hierarchical process models and can be described by widely accepted SCOR model[22][28].

- A Star/Consortia topology: Main contractor driven project consortia (hub-and-spoke topology of a network). Contractual issues play an important role. This type of VO is characterized by more stable project teams, which are coordinated by one project leader (main contractor), who has administrative and financial power [28].

- A Peer-to-peer topology: Project-based networks are example of this topology. These types of VOs are quickly re-assembled project organizations, which have a peer-to-peer topology of the network [28].

In the literature there are three generic types of accounts on VOs [22]. The first one is on organizations that extend some of their organizational activities externally, thus forming virtual alliances to achieve organizational objectives. VOs may be formed by integrating several companies' core competencies and resources [51]. In fact, the coordination of these business activities among organizations relies extensively on IT applications. The virtual corporation is then described as a network of independent companies - suppliers, customers, and even rivals - linked by IT to share skills, costs and access to another's markets. The second description of the VO is related to a perceptual organization that is "abstract, unseeing and existing within the minds of those who form a particular organization" [52]. The framework of VOs is often subjective and is open to many different perceptual interpretations. The VO is thus the antithesis of the physical organization with which we are familiar. This account explains how organizations are conceived and seen through their members. The third type of description is of organizations that are established with IT such as corporations with an intensive use of telecommuting.

No matter what VOs are striving to achieve they have some common traits. VOs provide distributed access across the space and time. Structure and processes running a VO are dynamic. Email, video conferencing, telepresence, awareness, social computing and group management tools are used to enable collaboration among the participants [10]. Operational organizations are supported by simulations, databases and analytical services. In daily life, we come across many VOs in terms of social

[22] http://supply-chain.org/resources/scor

networks as well (e.g., Facebook, MySpace). All these descriptions provide partial information about VO. A holistic framework for characterizing and studying VO is still missing.

2.1.6 Examples

A VO can provide solutions to different problem. It is difficult to specify or restrict the domain for which they are serving. Some advantageous roles played by VOs are,

- facilitator of access (BIRN [53], LEAD [54], NANOHUB [55]),
- enabler of system level science (SCEC [56], caBig [57], Large Hardon Collider [58]),
- enhancer of Problem Solving Processes (TeraGrid [48]) and
- key to Competitiveness (GEON [59]).

VOs have served in the field of earthquake engineering (The Southern California Earthquake Center) [56], cancer research (The Cancer Biomedical Informatics Grid (caBIG) [57]), climate research (The Earth System Grid [60]), high-energy physics (The Large Hadron Collider [58]), and computer science. Other communities are now forming VOs to study system-level science. These VOs and others are addressing problems that are too large and complex for any individual or institution to tackle alone. It simply is not possible to assemble at a single location all of the expertise required to design a modern accelerator, understands cancer, or predict the likelihood of future earthquakes. VOs allow humanity to tackle previously intractable problems [10].

2.2 Reference Architecture (RA)

An RA captures the essence of the architecture of a collection of system. The purpose of a RA is to provide guidance for the development of architectures for new versions of the system or extended systems and product families [61]. This section details definitions and examples of RA.

2.2.1 Architecture

An architecture is an abstract description of a specific system, i.e. a particular model that even at a logical level tends to indicate the system structure, functions of its components, their interactions, and constraints, and can be used to develop the system. Architecture is focused on "building a system" and must be complete at its level of abstraction; therefore not all models are architectures. The IEEE Standard 1471-2000 [62] defines an Architecture by the recommended practice as the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.

2.2.2 RA Definition

At the meeting of System Architecture Forum (SAF), following important questions were posed to understand the Reference Architecture

- *What is an RA?*

  Different definitions exist to answer this question. The one which is important in our context is presented by IEEE standard making institute. An RA is defined as a way of documenting good architecture design practice to address commonly occurring problem [62]. It is a way of recording a specific body of knowledge, with the purpose of making it available for further practical reuse[23]. A relevant source to better explain and understand these concepts is the work of the Organization for the Advancement of Structured Information Standards (OASIS) Service Oriented Architecture (SOA) Technical Group[24]. OASIS[25] is now a days key reference for the "development, convergence and adoption of open standards for the global information society".

- *Why do we need RAs, What is their value, What is the benefit of creating and maintaining them?*

  RA has specific *Vision, Goal, Benefits*, well defined in the boundaries of the scope. In this scope the RA links to relevant standards, legislation, domain constraints and mandatory frameworks [61] A very basic idea is to facilitate the intended domain users with a tool to achieve optimization by reusing the existing patterns. The value of RAs is foreseen in environments with a high multiplicity factor, creating social, organizational, business, application and technical complexity [61].

- *How do you capture a Reference architecture, How do you visualize it, What is the appropriate level of abstraction, How is it used?*

  An RA captures previous experience, for instance by mining, or by generalizing existing architectures. To be of value for future architectures, a Reference Architecture is based on proven concepts. The validation of concepts in Reference Architectures is often derived from preceding architectures.

OASIS, TOGAF[26], SHAMAN[27], ZACHMAN[28], NEXOF are the examples of valuable work done in the domain of Reference Architecture. The above listed questions are answered in Chapter 3 in the context of our proposed reference architecture in detail. For general information, in the context of SAF, is available at [61].

2.2.3 Criteria for a Good RA

Criteria for a good RA, as described by Gerrit [61] are:

[23]http://shaman-ip.eu/ (European Commission, ICT-216736)
[24]http://www.oasis-open.org/committees/tc_home.php?wg_aabrev=soa-rm/
[25]http://www.oasis-open.org
[26]http://www.opengroup.org/togaf/
[27]http://shaman-ip.eu/(EuropeanCommission,ICT-216736)
[28]http://zachmaninternational.com/2/Zachman_Framework.asp

- Understandable for a broad set of heterogeneous stakeholders (customers, product managers, project managers, engineers et cetera)
- Accessible and actually read/seen by majority of the organization
- Addresses the key issues of the specific domain
- Acceptable
- Up-to-date and maintainable
- Adds value to the business

2.3 Service Oriented Architecture

A *service* is defined as a function that is well-defined, self-contained, and does not depend on the context or state of other services [63]. Service Oriented Architecture (SOA) speaks of a collection of services, which communicate with each other, e.g., simple data passing or two or more services coordinating an activity. The SOA services follow the pattern of *publish*, *find* and *use*. The services are published through registration so that other services or users can discover them. After the discovery of a service, that service is contacted and then can be used. Service Oriented Computing (SOC) is becoming rapidly popular with an objective to change the life of individual, organizations and society in a similar way as the internet and the Web have done in the past decade. The SOC pledges the revolution of the Internet by a novel and advanced support for collaboration.

SOA facilitates the creation of flexible, reusable assets for enabling end-to-end SOA-based business solutions. The usage of the SOA-RA is a key enabler for the achievement of the value propositions of a SOA[29]. Goal of the SOA-RA is to provide a blueprint for creating or evaluating an architecture. Additionally, it provides patterns and insights for integrating these fundamental elements of an SOA as exemplified in the layers of an SOA [64]. Informally, the SOA-RA is designed to answer some of the key questions and issues encountered by architects, as detailed in [64], such as:

- *What are the aspects of an SOA as expressed in terms of layers that are important in designing solutions based on service oriented principles?*
- *What are the building blocks that must be included in each layer of solution?*
- *What are some of the key architectural decisions that must be considered to make when designing a solution that is based on a SOA?*
- *Which roles in a project would benefit from using these principles and guidelines?*

[29] http://www.opengroup.org/projects/soa-ref-arch

The SOA RA is used as a blueprint and includes templates and guidelines for architects. These templates facilitates and ultimately enable automation and streamlining the process of modeling and documenting the architectural layers, the Architectural Building Blocks (ABB) within them, options for layers and ABBs, mapping of products to the ABBs, architectural and design decisions that contribute to the creation of a SOA [64]. It is intended to support organizations adopting SOA, product vendors building SOA infrastructure components, integrators engaged in the building of SOA solutions and standards bodies engaged in the specifications for SOA [64].

2.4 Computing Paradigms

Computing paradigms and hardware technology compliment each other. Since the birth of networking and introduction of Internet, many computing and communication models have been developed and deployed. Major themes of collaboration kept on revolving around two main concepts, centralized and decentralized approaches. With the pervasiveness of technology, the vision of utilizing hardware and software resources as utility has become a reality. Grid computing and cloud computing are two remarkable paradigms to achieve utility computing. In the following sections we detail these two paradigms in detail. There are several other computing paradigms that existed for short or long periods but they are not in scope of this thesis.

2.5 Cloud Computing

The increased degree of connectivity and the increased amount of data has led many providers and in particular data centers to employ larger infrastructures with dynamic load and access balancing [16]. Term "cloud" appeared in 90s to refer the dynamic capability of traffic switching to balance utilization (telecom clouds) and to indicate that the telecom infrastructure is virtualized [65]. In 2001 Microsoft adopted this term in a public presentation about the .Net framework to refer to the infrastructure of computers that make up the Internet [66]. Cloud computing and its current understanding came into lime light when Amazon published Elasticity Compute Clouds in 2006 [67]. Multiple definitions exist, according to the context and capabilities.

According to Foster, "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet" [68]. For Gartner it is "a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external cusotmers using internet technologies" [15]. Cloud computing gain popularity in October 2007 when Google and IBM announced the "Blue Cloud" effort [69] [70].

Cloud computing abstracts the future of computing where computing is shifted from local to global platform, making third-party responsible for the provision of hardware and software resources as utilities. Thus supporting John McCarthy's prediction that "computation may someday be organized as a public utility".

2.5.1 Types

Types of cloud presented in "Cloud Computing Use Case"[30] by National Institute of Standards and Technology (NIST)[31] according to Deployment Models are:

- "Public Cloud: In simple terms, public cloud services are characterized as being available to clients from a third party service provider via the Internet. The term "public" does not always mean free, even though it can be free or fairly inexpensive to use. A public cloud does not mean that a user's data is publically visible; public cloud vendors typically provide an access control mechanism for their users. Public clouds provide an elastic, cost effective means to deploy solutions."

- "Private Cloud: A private cloud offers many of the benefits of a public cloud computing environment, such as being elastic and service based. The difference between a private cloud and a public cloud is that in a private cloud-based service, data and processes are managed within the organization without the restrictions of network bandwidth, security exposures and legal requirements that using public cloud services might entail. In addition, private cloud services offer the provider and the user greater control of the cloud infrastructure, improving security and resiliency because user access and the networks used are restricted and designated."

- "Community Cloud: A community cloud is controlled and used by a group of organizations that have shared interests, such as specific security requirements or a common mission. The members of the community share access to the data and applications in the cloud."

- Hybrid Cloud: A hybrid cloud is a combination of a public and private cloud that interoperates. In this model users typically outsource nonbusiness- critical information and processing to the public cloud, while keeping business-critical services and data in their control."

2.5.2 Everything as a Service (XaaS)

Research community is updating frequently with detailed fundamental and advance concepts about this newly evolved paradigm. Fundamental aspects described here are borrowed from the literature [16] [68] [71] [72] [73]. Cloud computing focuses on maintaining transparency between users and the computing details, thereby providing freedom to the providers to deliver IT services. It allows provider to manage cost, systems and Quality to suite the consumers and need for the business model being used [15]. This is the main target of service delivery. Once this has been

[30] http://opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf
[31] http://www.nist.gov/itl/cloud.cfm

achieved, the next level for many companies is to analyze, how many IT capabilities can be delivered as a service. Everything potentially becomes a service [15].

2.5.3 Cloud Stack

Cloud stack based on Everything as a Service (XaaS) is presented differently by research and business community. XaaS refers to X as a service architecture where X can be interpreted as anything, everything or all. XaaS is based on the concept of virtualization. Other popular XaaS types are Hardware as a Service (HaaS), Communication as a Service (CaaS), Network as a Service (NaaS), Component as a Service (CaaS) [16], Storage-as-a-Service [16], Human-as-a-Service [72]. Most agreed upon components found in literature are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [16] [72] [68]. These three most popular of XaaS type of service are grouped in the SPI model. These three concepts are detailed in the following section.

2.5.4 Software as a Service (SaaS)

SaaS is defined as a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own Web servers or upload the application to the consumer device, disabling it after use or after the on-demand contract expires. The on-demand function may be handled internally to share licenses within a company or by a third-party Application Service Provider (ASP) sharing licenses between companies (e.g. Google Docs, Salesforce CRM, SAP Business by Design) [16].

Alaxendar et al describes that all the applications that run on the cloud and provide a direct service to the customer are located in the SaaS layer. The application developers can either use the PaaS layer to develop and run their applications or directly use the IaaS infrastructure [72]. He distinguish between Basic Application Services (OpenId[32]) and Composite Application Services(Google Map[33]). Composite Application Service category include mash-up support systems with Opensocial as the prominent example allowing entire social networks like MySpace to be used as basic services [72].

2.5.5 Platform as a Service (PaaS)

PaaS is defined as the delivery of a computing platform and solution stack as a service. It often goes further with the provision of a software development platform that is designed for *cloud computing* at top of the cloud stack. It provides computational resources via a platform upon which applications and services can be developed and hosted (e.g. Force.com, Google App Engine, Windows Azure Platform) [16]. Platform offering provide an infrastructure for developing and operating Web based

[32] http://openid.net/foundation/
[33] http://code.google.com/apis/maps/index.html

software applications [71](e.g facilities for application design, application development, testing, deployment, and hosting, as well as application services such as team collaboration, security, application versioning and application instrumentation).

Alexandar [72] categorizes the services into *Programming Environments* and *Execution Environments*. Example of the former is Sun's project Caroline [74] and the Django framework [75], and examples of the latter are Google's App Engine [76], Joyent's Reasonably Smart [77] and Microsoft's Azure [78]. As seen by these examples an Execution Environment PaaS typically also encompasses a Programming Environment PaaS. One could potentially replace the Django framework in Google App Engine with her own Programming Environment and Microsoft Azure offers a wide range of alternative programming tools under the Azure runtime umbrella. This decoupling between execution and development environments is thus represented by having two categories in stack model presented in [72].

2.5.6 Infrastructure as a Service (IaaS)

IaaS is sometimes considered to be the provision of computer infrastructure (typically a platform visualization environment) as a service [79]. Alexender et al [72] divides IaaS layer in their proposed cloud stack into two parts. On the lowest level of the infrastructure closest to the hardware two types of services, Physical Resource Set (PRS) and Virtual Resource Set (VRS) services, are distinguished.

Geoffrey defines IaaS in a cloud stack as an infrastructure that provides distributed multiple physical components to support cloud computing, such as storage and processing resources. This layer allows the infrastructure to abstract away details such as which exact hardware an applications is using and which data center the application is running in [71]. Virtual Machine (VM) concepts have also enabled this transparency between hardware implementation details and providers thereby increasing the ability to rapidly scale server resources in response to changing demand [71] IaaS also referred to as *resource clouds*, provide (managed and scalable) resources as services to the user - in other words, they basically provide enhanced virtualization capabilities [16]. Accordingly, different resources may be provided via a service interface: *data* and *storage clouds* deal with reliable access to data of potentially dynamic size, weighing resource usage with access requirements and/or quality definition. IaaS offers additional capabilities over a simple compute service. Examples: Amazon EC2, Zimory, Elastichosts, Secure Storage Service (S3) [16].

2.6 Grid Computing

The term *grid* is chosen as an analogy to the electric power grid that provides consistent, pervasive, dependable and transparent access to electricity, irrespective of its source. In mid 1990s the computer scientists began exploring the design and development of analogous infrastructure called "Computational Power Grid". Since then many definitions of grid has been launched. Most famous are listed here.

Carl Kesselman and Ian Foster, in 1998 wrote in their book "The Grid : Blueprint for a New Computing Infrastructure" [80], "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."

In a subsequent article, "The Anatomy of the Grid", authors with Steve Tuecke in 2000 [81], refined the definition to address social and policy issues, stating that grid computing is concerned with "coordinated resource sharing and problem solving in dynamic, multi-institutional VOs". The key concept is the ability to negotiate resource-sharing arrangements among a set of participating parties (providers and consumers) and then to use the resulting resource pool for some purpose. According to Authors [81],

"The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, soft ware, data, and other resources, as is required by a range of collaborative problem solving and resource brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a VO. We also spoke to the importance of standard protocols as a mean of enabling interoperability and common infrastructure".

2.6.1 A Grid Checklist

Foster's three point checklist of a grid system [82] present a clear vision, according to which a *grid* is a system that:

- "Coordinates resources that are not subject to centralized control. A grid integrates and coordinates resources and users that live within different control domains-for example, the user's desktop vs central computing; different administrative units of the same company; or different companies; and addresses the issues of security, policy, payment, membership, and so forth that arise in these settings. Otherwise, we are dealing with a local management system".

- "Using standard, open, general-purpose protocols and interfaces. A grid is built from multi-purpose protocols and interfaces that address such fundamental issues as authentication, authorization, resource discovery, and resource access. As I discuss further below, it is important that these protocols and interfaces be standard and open. Otherwise, we are dealing with an application specific system".

- "To deliver nontrivial qualities of service. A grid allows its constituent resources to be used in a coordinated fashion to deliver various qualities of service, relating for example to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than that of the sum of its parts".

According to Rajkumar Buya, "A grid enables the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organizations for solving large-scale resource intensive problems in science, engineering, and commerce" [83].

Grids have moved from the obscurely academic to the highly popular. We read about compute grids, data grids, science grids, access grids, knowledge grids, bio grids, sensor grids, cluster grids, campus grids, tera grids, and commodity grids. Ian Foster and others posited that by standardizing the protocols used to request computing power, we could spur the creation of a computing grid, analogous in form and utility to the electric power grid. Researchers subsequently adopted the idea and materialized it by producing for example large-scale federated systems [68]. Examples include TeraGrid [48], Open Science Grid [49], caBIG [57], EGEE [84], Earth System Grid [85], which just not only provide computing power, but also data and software, on demand. Open Grid Forum (OGF)[34], OASIS defined relevant standards for grid computing. More prosaically, the term was also co-opted by industry as a marketing term for clusters. But no viable commercial grid computing providers emerged, at least not until recently [68].Characteristics of grid computing coordinate independent resources, use open standards and interfaces, Quality of Service (QoS) allows for heterogeneity of computers distribution across large geographical boundaries loose coupling of computers.

2.6.2 Cloud vs Grid

Cloud computing is a concept rather a technology. Research community is facing an ambiguous situation about relationship between grid and cloud [86]. Sometimes clouds are seen on the top of grid and vice versa or even identical. Many researchers put their effort to differentiate them in elaborated comparisons [87] [88] [89] [68] but still have different views on what "the grid" is in the first instance, making it more confusing. We have to look inside what grid is carefully. Grid in core, like cloud, is a concept rather than a technology.

Authors mention to distinguish between (1) Resource grids including in particular grid computing and (2) e-Business grids which centers mainly on distributed VOs and is closer to SOAs [16]. There may be combination between two, e.g. when capabilities of the e-Business applied for commercial resource provisioning, but this has little impact [16]. Resource grids and e-Business are distinguishable and each possess similarity to cloud computing in different parameters. An elaborated comparison presented in [90] [91] is summarized below:

Similarities

- Share the same goal of resource provision [90]
- Scalability [90]

[34] http://www.gridforum.org/

- Multitasking and Multi-tenancy [90]
- Service Level Agreements (SLAs), as both paradigms need resources from a third party (cloud vendor or collaborator in grid) [90].

Differences

- Grid computing stems from academia or more precisely the field of High Performance Computing (HPC) while cloud computing stems from industry [91] [90].
- Grid supports heterogeneity of resources while cloud computing is centralized [90] [91].
- Grid posses no specific Business Model while cloud operates on Pay-as-you-go.
- Application Developers: Developing application on grid requires the exhaustive knowledge of the grid environment. In cloud environment, for IaaS, developers can customize their working environment with their familiar tools and configuration, similar to working on their own local machines[91]. For PaaS, service provider supplies a platform SDK (Software Development Kit) and/or some debugging tool (e.g. Google App Engine, Google plug-in for Eclipse) [91].
- Running a task: In grid, end user has to specify the type and quantity of resources desired, authentication information, program to be run and its parameters, sources of input and output and its destination. For example, globusrun-ws[35], the command supplied by GT4[36] for job submission and management, has 30 options for submitting a job and 15 options for monitoring a job. The risk of making an error is higher and it requires expertise on end-user's part [91]. Also, the grid middleware, being a software itself, has its requirements on the running environment. Existing grids are tightly platform dependent. For example, gLite[37] presently can only run on Scientific Linux 4 and 5, and Debian[38] 4. Cloud computing, as a contrast, make the job submission and execution easier through Virtual Machines (VM) technology. Only requirement is to reserve the resources and configure them with several mouse click [90]. Constraints laid by grids on the running programs, no longer exist in cloud [91].
- Cloud computing is used to host web services that tend to run for long time (long-serving daemon-like services) as opposed to grid applications that tend to be more compute intensive and batch-like [90].

The challenge what we have to address here is how to move from VOs of grid environment to virtualization of cloud environment. By principle, in grids the resources are not subjected to centralized control hence the concept of VO has been introduced. VO refers to "a group of individuals and/or institutions and resources that

[35] http://www.globus.org/toolkit/docs/4.0/execution/wsgram/rn01re01.html
[36] http://globus.org/toolkit/docs/4.0/
[37] http://glite.cern.ch/
[38] http://www.debian.org/

work in collaboration towards a common goal". The users of grid can be organized in different VOs each having different set of policies. The authorization in grid is at the VO level, i.e. a user belonging to a particular VO can access those resources that supported by her VO. Hence VO in grid unifies the resources belonging to different administrative domains. In a gLite[39] based grid environment, the grid sites can choose which VOs to support at what level by the administrator. Hence the user at a grid site can join one or more VOs supported by the site by passing through the required authentication and authorization procedure.

2.6.3 Grid, Cloud and SOA

The technical report "Beyond Cloud Computing" published in 2009 details relationship between grid, cloud and SOA. We present extract from the report [16].

"There is a strong relationship between the "grid" and "SOAs", often leading to confusions where the two terms either are used indistinguishably, or the one as building on top of the other. This arises mostly from the fact that both concepts tend to cover a comparatively wide scope of issues, i.e. the term being used a bit ambiguously. SOA however typically focuses predominantly on ways of developing, publishing and integrating application logic and/or resources as services. Aspects related to enhancing the provisioning model, e.g. through secure communication channels, QoS guaranteed maintenance of services etc. come in this definition secondary. Again it must be stressed though that the aspects of e-Business grids and SOA are used almost interchangeably - in particular since the advent of Web Service technologies such as the .NET Framework and Globus Toolkit 4, where GT4 is typically regarded as grid related and .NET as a Web Service/SOA framework (even though they share the same main capabilities)".

"Though providing cloud hosted applications as a service is an implicit aspect of cloud Software as a Service (SaaS) provisioning, the cloud concept is principally technology agnostic, but it is generally recommended to build on service-oriented principles. However, in particular with the resource virtualization aspect of cloud systems, most technological aspects will have to be addressed at a lower level than the service layer. SOAs are therefore of primary interest for:

- the type of applications and services the user can build for and host on the cloud system.
- for providing additional high-level.

services and capabilities with which to enhance the base cloud capabilities".

Geoffrey Raines explains relationship between cloud computing and SOA in [71]. According to him "Service orientation is an easy approach to bring good, from NEXOF, SOA is not a specific technology or predefined solution but rather a paradigm or architectural style that is used to improve the scalability and decentralization within distributed and heterogeneous IT environments. This is important since processes and systems are becoming more and more complex and IT landscapes

[39]http://glite.cern.ch/

are rapidly changing. SOA aims at closing the gap between business and IT in these environments in order to flexibly and efficiently exploit business opportunities. Like all other software system architectures, SOA is a non-tangible characteristic of a software system that can be captured within models, specifications and accompanying material. Based on these specifications a concrete implementation can be built".

2.7 Summary of Research Contribution

This chapter explained the VO, Reference Architecture and related terminologies in detail. These concepts are baseline to our research efforts. Next chapter details the proposed RAVO in detail.

3 Reference Architecture for Virtual Organization (RAVO)

3.1 Motivation

Information Technology (IT) has become an essential part of our daily life. Utilization of electronic platforms to solve logical and physical problems is extensive. Grid computing is often related with VO when it comes to creation of an E-collaboration. The layered architecture for grid computing has remained ideal for VOs. Example success stories include LEAD [54], NANOHUB [55]), (SCEC [56], caBig [57], Large Hardon Collider [58]), TeraGrid [48]), GEON [59]. Ranging from the field of earthquake engineering (The Southern California Earthquake Center) [56], cancer research (The Cancer Biomedical Informatics Grid (caBIG) [57]), climate research (The Earth System Grid [60])to high-energy physics (The Large Hadron Collider [58]) VOs are serving humanity.

However, grid computing paradigm has some limitations. Existing grid environments are categorized as *data grid* or *computational grid*. Today, problems being solved using VOs require both data and storage resources simultaneously. Scalability and dynamic nature of the problem solving environment is another serious concern. Grid computing environments are not very flexible to allow the participant entities enter and leave the trust. Cloud computing seems to be a promising solution to these issues. On-fly, demand driven, scalable and dynamic problem solving environments are target of this newborn approach. Cloud computing is not a deviation concept from the existing technological paradigms, rather it is an upgradation. Cloud computing centers around the concept of XaaS, ranging from hardware/software, infrastructure, platform, applications and even humans are configured as a service. Most popular service types are IaaS, PaaS and SaaS. We have described these concepts in detail in Chapter 2.

Keeping in view the current technological and computing paradigms evolution, changed requirements and lack of standards for VO, we propose an RA for creating a VO. The proposed RA also integrates these concepts and reflects the need of providing data and computational resources by supporting all types of VOs. This chapter presents the existing standards followed by our proposed standard. We detail the requirement analysis and component identification phase regarding which is must for the participating entities to clear their concepts and to build a trust. Also, it establishes the basis for must and optional components of the VO. , proposed RA in detail. It explains a layered architecture to combine the building blocks and define the relationship between them during a problem solving activity. Another achievement regarding stakeholder of a VO is the introduction of a new concept

Subject. Stakeholder and resources in a VO are closely related. Chapter 4 details this part of the RAVO.

3.2 Existing Frameworks/ Efforts done

Efforts have been continued to standardize the VO but unfortunately there are is no RA, specific to VO, available to date. Different models have been developed and theoretical discussions enlightened various important research issues regarding VOs. Major focus remained on VE and to achieve more efficient e-Business platforms. Also, work done is domain specific and it is rare to find a generalized approach followed by other researchers, even if it existed. We discuss prominent and related efforts in this section.

3.2.1 NEXOF

The main goal of the NEXOF-RA[1] project is to provide an RA for service-based software systems which facilitate the reuse of well proven Service-Oriented concepts. The NEXOF RA focuses on the architecture of a service-based software system infrastructure. It is provided in form of a construction kit that guides the construction of specific SOA infrastructures. The construction kit consists of a set of building blocks implementing architectural patterns. These architectural patterns in turn are related to a conceptual architecture model.

The NEXOF RA model captures the relevant entities and concepts on a conceptual level as well as their dependencies that constitute such a Service-Oriented system. The NEXOF RA Model fosters the communication about the relevant elements on a higher abstraction level. A SOA [92] based solution provides (amongst other things) an infrastructure on which services can be deployed and executed in a distributed system. The NEXOF-RA project describes a RA for distributed systems. The project only addresses the architecture of the infrastructure [79]. Concrete applications and services are not in the focus since the RA should be domain independent and open. The infrastructure architecture addresses the hardware infrastructure architecture as well as the software infrastructure architecture. Some basic services are also provided by the overall infrastructure in order to allow the operative elements to be exploited. Thus, this infrastructure, the NEXOF-RA infrastructure can be perceived as an operating system for services and Service-Oriented applications.

3.2.2 SHAMAN

The SHAMAN[2] RA is based on the OASIS SOA Reference Model. The SHAMAN project has three aims: the development of a next-generation digital preservation framework; the development of the corresponding digital preservation tools; and the development of a RA of evolving nature. The development of the SHAMAN RA follows the definitions and guidelines provided by [62], concepts and relationships

[1]http://www.nexof-ra.eu/?q=node/330
[2]http://shaman-ip.eu/shaman/

described in[3], and is based on the version 9 of The Open Group Architecture Framework[4] (TOGAF), Enterprize Architecture (EA) framework. Since the acceptance of the SHAMAN Reference Architecture will depend on the traceability of the proposals to well understood requirements and design decisions, it is necessary to consider the three domains of focus and their related implementations. SHAMAN identifies high-level features required by a preservation system. These include those found in the OAIS, as well as additional features not present in OAIS [93]:

- "Layered Information Package where each layer is addressed by a particular preservation activity".
- "Refinement of the Information Package is needed to ensure that the information necessary to guarantee long-term preservation is included".
- "Activities that precede the Ingest (Pre-Ingest) need to be investigated and their contributions and impact on digital preservation need to be evaluated".
- "Activities that succeed the Access (Post-Access) need to be investigated and their contributions and impact on digital preservation need to be evaluated".

The SHAMAN RA requires detailed understanding of these items and the OAIS model which can only come from more detailed investigations and from interaction with the preservation community. To account for this, SHAMAN has put in place an iterative approach where information from Integrated Sub-Projects and the community is used to drive further elaboration of the Reference Architecture [93].

The SHAMAN RA is an EA-based approach that enables the accommodation of digital preservation concerns in the overall architecture of an organization. For that, a capability-based model of preservation based on established digital preservation key references and best-practices from related fields was derived from research on the stakeholders of the domain, their concerns, goals, and influencers (drivers and constraints). The result is a general understanding of the domain, providing a multidimensional view on the concepts covered on these key references. The approach taken with this reference architecture enables the transfer of Digital Preservation (DP) know-how into a non-traditional repository-based DP scenario, since it is itself agnostic to concrete scenarios. In other words, this capability-based approach can deliver value to organizations in which the preservation of contents is not a main business requirement, but required to enable actual delivery of value in the primary business.

SHAMAN RA is specific for the preservation of digital objects. Its targets preservation community only, although it is based on EA, SOA and OASIS. We need a standard which addresses the creation of VO for any single or combination of domains.

[3] http://www.omg.org/spec/BMM/1.1/
[4] http://www.opengroup.org/togaf/

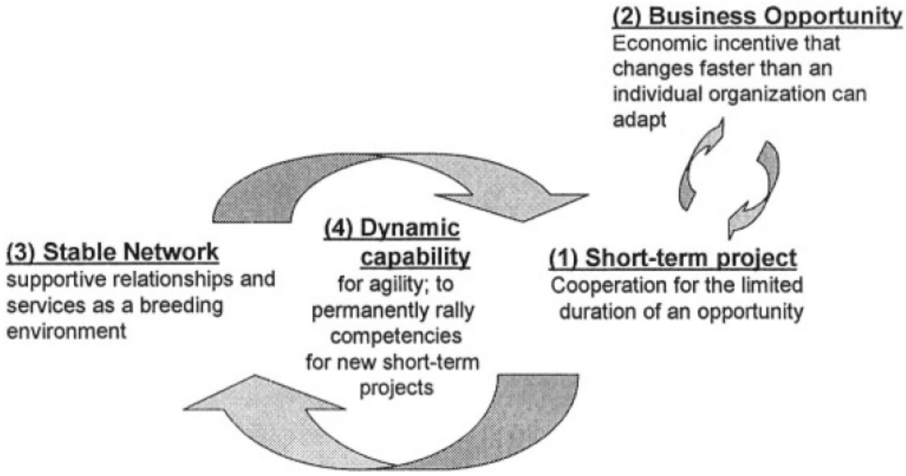

Figure 3.1: *Reference Model for Virtual Organizations[7]*

3.2.3 Reference Model for VOs

Katzy et al carried out a study of organizational patterns across 20 projects that could be an early descriptions of possible types of VOs at the beginning of the century [7]. They investigated information systems, and more important, coordination roles, network structure, and strategies as complementary elements of a consistent structure and proposed three distinct basic types of VOs: *Supply chain*, *Lead contractor*, and *Peer projects*. Their work highlight network types, management roles, business opportunities and life span of VO as important aspects of a VO. Proposed Model is shown in Figure 3.1

3.2.4 A Reference Model for Collaborative Networked Organization (CNO)

Luis M Camarinha-Matos and Hamideh Afsaemanesh presented a generic conceptual model that synthesizes and formalizes the base concepts, principles and recommended practice for collaborative networked organizations. It provides a guided path to facilitate the creation of focused models for different manifestations of CNO's as well as architectures and implementation models for particular system developments. It provides basis for the derivation of models closer (not directly) to the concrete case. The model is named ARCON (A Reference Model for Collaborative Networks) and it was developed in ECOLEAD[5] project. It claims to provide a holistic approach combining technology and business prospect. It also addresses the culture, values, norms and principles, trust as dimensions of the proposed model. Model is shown in Figure 3.2, more details about this model are available at [8].

3.2.5 VOSTER

European Union (EU) funded project names VOSTER (Dec2001-May2004) was dedicated to collect, analyze, synthesize the result from a number of leading European

[5] http://www.ve-forum.org

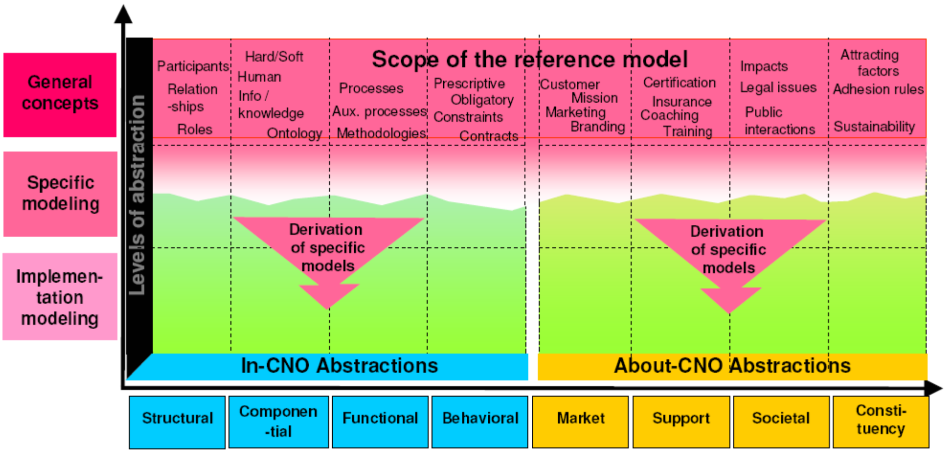

Figure 3.2: *A Reference Model for Collaborative Networked Organization[8]*

research projects on VO. It aims to consolidate VO reference models and related modeling methodologies based on experiences acquired in thirty relevant EU funded research projects [24]. The outcome of the project is not a model or architecture rather it presented open questions to the research community. The research reveals the complex reality of deployment and adoption of VO practices and identifies a number of organizational, legal, economic, socio-cultural, and technical challenges faced by VOs [24]. Aisha Abuelmaatti and Yacine Rezgui has elaborated the characteristics of virtual business models and suggested that the value-added alliance equation consists of a combination of technology, organizational, and ultimately legal and economic considerations. Thus, in researching, developing, and evaluating potential VO solutions, these issues must be blended successfully toward the shared VO purpose [24]. Given current limitations of VOs research, the contribution is made to existing knowledge by raising a number of research questions related to

- Clarifying and defining the nature of virtual business modes that takes place amongst organizations,

- Specifying the technological, regulatory and socio-organizational environment to support VOs effectively; and

- Researching into factors that facilitate virtual business modes adoption and use across organizations.

Emphasis is given on further research in technology maturity and software provision models, organizational and process settings, and social, including socio-emotional considerations, adapted to the needs of organizations. Authors highlights the case for the need to develop a business oriented social and organizational road map, aimed not only at senior management but all categories of an organization's staff [24].

3.2.6 Comparison

Research efforts done to standardize VO building process could not provide any common agreed upon framework. The above detailed efforts pointed out important issues related to standardization process. We compare these efforts with our research focus. First, there is a difference between Reference Model and RA.

A *reference model* is a generic abstract representation for understanding the entities and the significant relationships among those entities of some area, and for the derivation of other specific models for particular cases in that area. Preferably a reference model is based on a small number of unifying concepts and may be used for education, explaining purposes, and systems' development [8].

Whereas, *RA* aims at structuring the design of architectures for a given domain by defining a unified terminology, describing the functionality and roles of components, providing template components, giving example architectures, and defining a development methodology [8]. It corresponds to architecture as a style or method in the sense that may represent a coherent set of design principles to be used in a specific area. The RA is the basis for designing the specific architectures for particular instances of systems in the class of systems covered by the RA [8].

In the Collaborative Networked Organization's domain, a RA for VO management systems would represent the "structure" and principles to be followed by particular architectures of concrete VO management systems. The concept of RA also induces the creation of generic re-usable "building blocks" [8].

Our proposed RA provides a structure with re-usable building blocks to create a VO in target domain. Layered architecture enables participating entities to collaborate at any required level. It is not necessary that a VO must have three layers (SaaS, Paas, IaaS). Requirements decide which layer to be skipped and vice versa. Answering a series of questions, defining components to be included sets the base for developing a concrete architecture by utilizing the best suited technology.

3.3 Reference Architecture for Virtual Organization (RAVO)

Building of RAVO is divided into two parts. We aim to develop a RA for the VO that integrates user's requirements and technology shift flexibly and dynamically. In first part, called *Requirement Analysis Phase*, we established theoretical grounds to justify the need of building a VO and identifying its components. This standard activity plan or pattern is applicable to any domain and to any type of VO. This pattern is verified by a pilot approach to evolve grid based Neural Network (NN) System N2Grid [94] in to Virtual Organization for Computational Intelligence (VOCI) [1] [3]. The second part details the generic architecture for RAVO. These parts are detailed in the following sections

3.4 Requirement Analysis Phase

The creation of a VO is time consuming and should be a well planned activity. In this section we will discuss VO and technology from different perspectives. Both aspects are required to support each other. Technology provides the basic infrastructure for a VO to exist. A VO in turn places demands on IT and shapes the evolution of technology. For the last decade the VO is one of the most discussed collaboration environment; but still there no standards exist.

From this discussions we assume a step wise approach which is helpful in the creation of VO. It can be separated in two phases which are detailed below:

3.4.1 Phase 1: Questions

The creation of VO starts with a series of questions, which are very critical in order to proceed. These questions (Qx) are listed in the following:

- Q1: Why to form a VO? What are the reasons of an organization to create a VO?
- Q2: What is the motivation behind participation? Why should other persons, institutes, service providers want to participate in a VO?
- Q3: What services are offered by a VO?
- Q4: How are these services fared? What is the type of the resources/business model?
- Q5: Who are the intended users? Who will eventually use and get benefited from this VO?
- Q6: What is the life of (membership of) a VO? Are temporal alliance or permanent participation expected?

3.4.2 Phase 2: Identification of Components

Based on these Q&A activity it is necessary to identify the building blocks of a VO. Gannon [95] has identified main components of a VO. These components are

- Common interest: The reason to form a VO.
- Users: the participants of a VO.
- Tools and services: This is a crucial part of a VO, which maintains the overall working environment and saves the existing patterns to be reused in order to reduce time to solve similar problems. A VO requires a collection of shared analysis tools (e.g. visualization tools and provenance tools). Tools can be integrated into specific VO work flows and can be shared and reused. They are used to curate data and publish results.
- Data: A VO contains two types of data, generally categorized as meta data and operational data that is being operated by tools.

3.5 RAVO: Generic Architecture

According to Gerrit Muller [61] there are two simultaneous trends,

- Increasing complexity, scope and size of the system of interest, its context and the organizations creating system [61].
- Increasing dynamics and integration: shorter time market, more interoperability, rapid changes and adaption in the field, in a highly competitive market, for example cost and performance pressure [61].

These trends form basis for our proposed RA as well. VOs are developed as distributed system at multiple locations, by multiple entities, consisting of multiple applications by multiple vendors, merging multiple domains for providing solutions to multiple problems. RA comes in scene where the multiplicity reaches a critical mass triggering a need to facilitate product creation and life cycle support in this distributed open world [61]. We detail the RAVO in the subsequent sections.

3.5.1 Definition

We define RAVO as "an open source template that does not only depict the architectural patterns and terminology, but also defines the boundaries where heterogeneous resources from different domains merge collaboratively into a common framework".

A RA has a life span and is dependent on the target architecture and possibly other RAs. As guideline for our effort we closely analyzed the RA presented by SHAMAN[6], GERRIT MULLER [61] and NEXOF[79]. RAVO provides

- A common lexicon and taxonomy.
- A common (Architectural) vision.
- Modularization and complementary context.
- A layered approach(bottom-up).

3.5.2 Goal

A common vision facilitates the participating entities to work as a team to achieve their decided goals. Modularization helps to integrate different domains thereby decreasing the efforts and context information make the dynamic nature of the architecture consistent.

We aim for developing a RA which allows for new forms of IT infrastructure coping with new collaborative processing paradigms, as grid computing and cloud computing. Thus we have to deliver an environment to allow for the new *Internet of Services and Things* accommodating the novel service stack, as IaaS, PaaS and SaaS. Architecture is classified into different layers according to the service each layer provides. Layered architecture is chosen because it helps to group different components (logical and physical) according to the degree of relatedness and required functionality.

[6] http://shaman-ip.eu/(EuropeanCommission,ICT-216736)

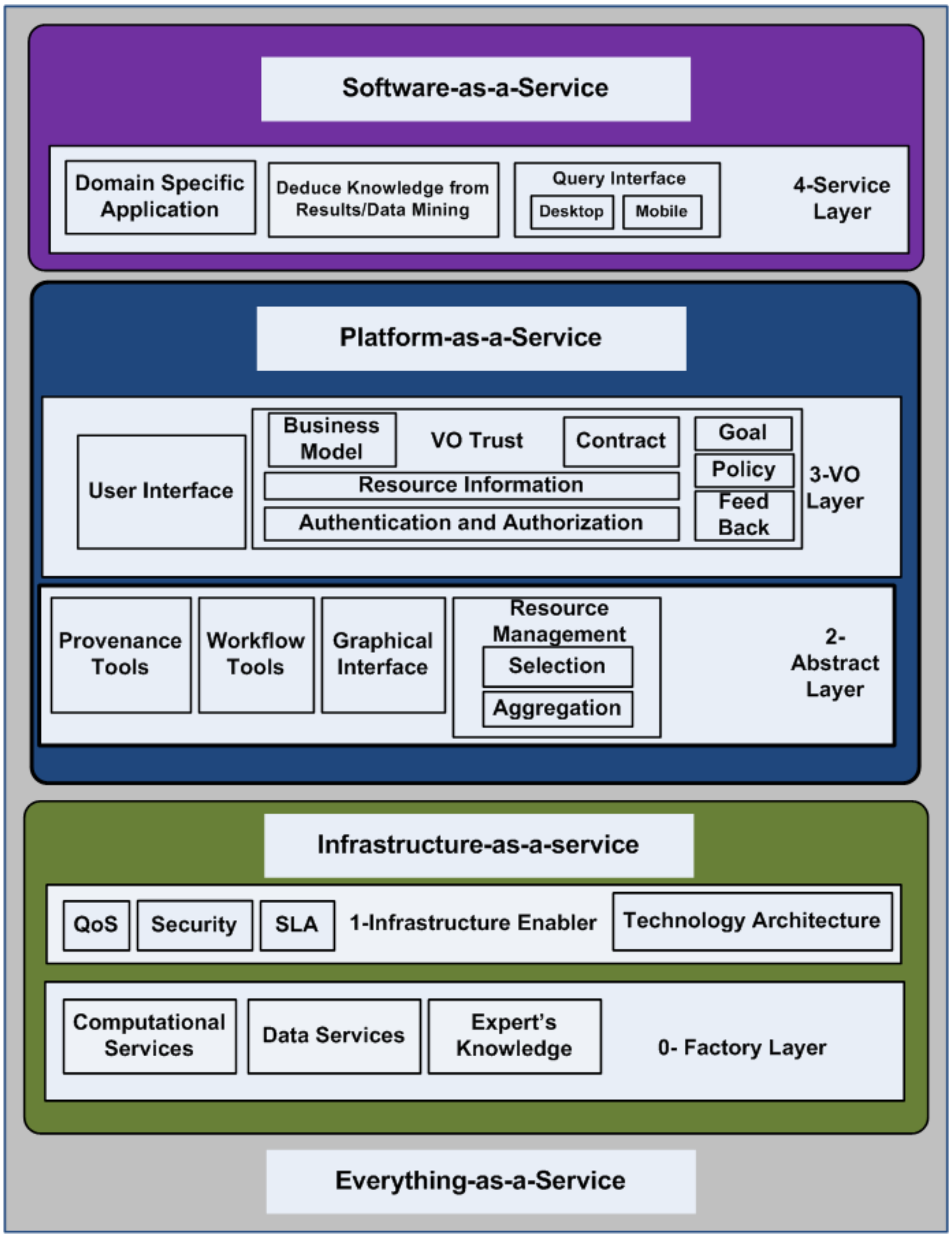

Figure 3.3: *XaaS Skeleton of RAVO*

3.5.3 Components and SPI based Framework

RAVO is based on SPI model. Layered approach is used to achieve the goal of providing all the resources as a service. Layers are distributed into 3 broad categories, IaaS, PaaS, SaaS

Figure 3.3 presents the framework for VO using the SPI model. The layers are distributed into 3 broad categories, IaaS, PaaS and SaaS thus resulting in XaaS.

3.5.3.1 Software as a Service Layer

In context of RAVO, SaaS is composed of a Service layer. It contains Domain Specific Applications (DSA) accessible by all users. DSAs are the combination of several user interfaces and Business Models found in the VO layer. Users, who only use the platform to solve their domain specific problems and do not contribute to the VO, find an entry point at this layer.

- *Service Layer*: It has open source, downloadable software, categorized in domains. The Service layer packages several services provided by the VO layer to be subscribable entities. These entities include generic functionality to query information from the problem domain as well as the means to perform data mining on the compound data created or provided by the combination of the services.

3.5.3.2 Platform as a Service Layer

In RAVO two layers, namely VO layer and Abstract layer, cover PaaS.

- *Abstract Layer*: This layer is composed of essential tools which enable the whole framework to be exploited in a dynamic manner. The set of tools consist of provenance, workflow, graphical tools and any other domain specific tools which are used to enhance the reuse of the resources for a diverse set of problem solving activity. Each tool provides its own functionality, its own user interface description [96], as well as an abstract API (identical for each tool) to access the resource in Factory layer.

- *VO Layer*: This layer is the entry point for user. It provides the realization of the user interface description and defines a business model on top of the Abstract layer to set usage cost according to usage statistics. Participating entities can agree on a usage model and build a cost trust for selling their resources. In context of VO, contributor/subject users (who not only use the resources offered by a VO but also contribute to the VO) are authorized to access the system on this layer. All have access to the system on PaaS layer.

3.5.3.3 Infrastructure as a Service Layer

In RAVO logical and physical resources are considered to be the part of IaaS. This part consists of two sub-layers in RAVO: Factory layer and Infrastructure Enabler layer. Only users with administrative rights have access to this layer.

- *Factory Layer*: Belongs to the IaaS category and contains resources for RAVO. Resources are described as physical and logical resources. Physical resources comprise of hardware devices for storage and computation cycles in a distributed manner. Logical resources contain expert's knowledge that supports the problem solution activity thereby reducing time to reach the specified goal.

- *Infrastructure Enabler Layer*: Allows access to the resources provided by the Factory layer. It consists of protocols, procedures and methods to contact the desired resources for a problem solving activity. It acts as a glue or medium to reach the desired resources based on user request.

3.5.3.4 Everything as a Service Layer

All layers are providing their functionality in a pure Service-Oriented manner so we can say that RAVO is XaaS.

3.5.4 Design Perspective of RAVO

VOs is a broad category of distributed systems. It is envisioned as a combined effort of multiple entities (organizations, people, HW, SW) for achieving a goal. Building RA for VO is effective in many ways. RAVO forms basis for VOs belonging to any domain. It improves the effectiveness by managing synergy, providing guidance for collaboration, generic framework, managing and sharing the architectural patterns. Interoperability is the most critical aspect of collaborative computing and VOs main feature. It determines the usability, performance and dependability of user level applications [61]. Integration cost and time are also important factors in context of interoperability. RAVO supports interoperability by defining a negotiation model/trust for the participating entities thereby supporting the effective re-use of patterns. Many RA focuses the technical architecture only. According to SAF meeting conclusion, A RA should address [61],

- Technical architecture.
- Business architecture.
- Customer context.

RAVO well addresses these three aspects. It presents a technical architecture specifying the must participating modules, APIs, protocols and platform to support VO. RAVO offers a Business Model which is open according to the participating entities conditions for resource sharing. Business Model and customer context overlap. RAVO explicitly defines roles of participating entities as *Subject*, *Consumer*, *Producer* and *administrators*. Elaboration of roles makes it easy to dynamically update the Business Model as an entity changes the role. RAVO supports feedback from the participating entities which is helpful in improving and maintaining the existing RA. These concepts are already detailed in RAVO section.

RA is a perceived image of existing technologies. Designing RA is a challenging job because it needs sufficient proof to justify its need in the said context. RAVO focuses on VOs. To the best of our knowledge, there is no standard pattern or framework which can be used to create a VO from scratch. Our vision is to provide the VO community a complete framework for identifying main components and abstract a life cycle to create VO from scratch. It grasps knowledge from existing structures

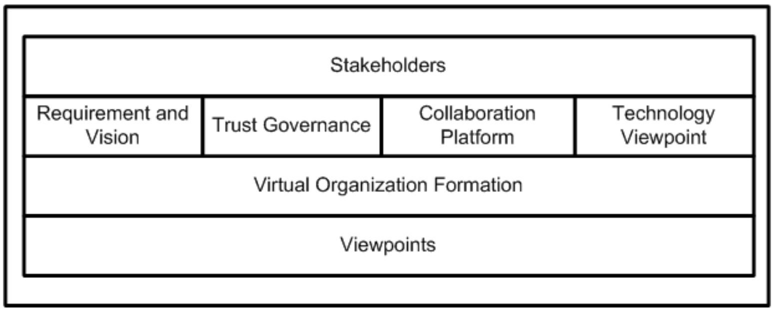

Figure 3.4: *Viewpoints in Reference Architecture for Virtual Organization*

such as NEXOF [79] and SHAMANs[7]. Guidelines are used to modify the requirements into an RA which supports creation, dynamic evolution and maintenance of a VO.

3.6 Viewpoint

Viewpoint is defined as a specification of the conventions for constructing and using a view. A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis [97]. View[8] is a representation or description of the entire system from a single perspective. Stakeholder is the viewer, who perceives the system according to her role. Viewpoint has a name, stakeholders addressed by it and concerns to be addressed by the viewpoint, and the language, modeling techniques or analytical methods to be used in constructing a view based on the viewpoint [97]. According to these definitions, viewpoints extracted from the concerns of the stakeholders are shown in Figure 3.4. These viewpoints are detailed in the following subsections.

3.6.1 Forming a Virtual Organization

Stakeholders collaborate to form a VO. All participants of a VO have an objective (personal or organizational) to achieve via this collaboration. Sub-viewpoints are,

- Domain definition: Depends on the type of problem solution, target domain can be one or multiple. Thus, stakeholders can be from one or multiple domains.
- Participation Level: Participation can be individual or at organizational.
- Duration: Stakeholder remain part of the VO according to the membership duration agreed upon among the collaborative entities. It can vary depending on the type of VO, partial or permanent (either participation is required for a specific part or throughout) and a Business Model in case of profitable organizations.
- Types of Contribution: It is decided by the role assigned to a specific stakeholder in the context of a VO. These concepts are detailed in Chapter 4.

[7] http://shaman-ip.eu/(EuropeanCommission,ICT-216736)
[8] http://shaman-ip.eu/(EuropeanCommission,ICT-216736)

3.6.2 Requirements and Vision

This viewpoint formulates the boundaries of a VO and its participants. All the participants must clearly put their requirement and goals while building collaboration. These requirements should reflect any assumption made on the architecture and the respective requirements stemming from the assumptions. Once requirements are defined (in form of a list, catalogue), VO has a vision to achieve and targets are set accordingly.

3.6.3 Trust Governance

Trust governance viewpoint is very important to any collaboration specially for VOs. Keeping stakeholders and resources glued together to achieve a target is only achievable via strong trust. Following sub-viewpoints are defined in this context:

- Trust/Policy formation: Experts and planners from participating entities prepare an agreed upon policy/model/contract. This policy defines the rules for participating and leaving VO, contributing and consuming resources, penalties for violation and measures to keep consistent and just to all the stakeholders.
- Objective Catalogue: This viewpoint provides the list of all the contracts and agreements in a documented form necessary for authentication, authorization and stakeholder management.
- Reviewing Policy: Due to dynamic nature of VO, the policies and contracts are reviewed to be in the accordance of change in requirements, technological updates, removal and entrance of participants.
- Business Perspective: This viewpoint is optional depending upon the type of VO. Profitable VO have a Business Model for metering, billing in addition to authentication and user management.

3.6.4 Collaboration Platform

This view point provides details of participating components for realizing the VO on technological and system level. It is further divided into 4 sub-viewpoints which are briefed here as,

- Data: This viewpoint aims to depict the types of data utilized in collaboration. Two broad identifications are found as *meta data* and *operational data*. Problem nature, domain and participating entities decide on the data source and security in collaborative efforts. Data and relationship between different components can be represented using Table, Mat, UML Class diagrams, Activity diagrams and Component diagrams.
- Applications and Tools: This viewpoint describes the list of running applications and tools utilized in a problem solving activity. This view can be further divided according to requirement. Roles of stakeholders also decide the access to

different available tools and applications at multiple levels (Interface, infrastructure, platform and so on). Distribution and relationship among applications, tools and components can be shown using UML Component diagram.

- Resources: This viewpoint explains the list of resources (Table, List), their owners, availability, usage cost (in case of profitable organization) and access rights. We have to sub-viewpoints:
  - Subject: An important viewpoint which defines stakeholder which consume and contribute to the resources simultaneously. Subject viewpoint is discussed in detail in Chapter 4.
  - Enabler: This viewpoint details the stakholders which are related to deployment, configuration, monitoring and lifestyle management. Roles assigned in this viewpoint are developers, administrators, business providers, planners and experts. Details are available in Chapter 4.
- Log catalogs: This viewpoint keep track of activities which are carried out during problem solving activities. Dynamic collaboration environments need to this record for the feedback and improvement.

3.6.5 Technology Viewpoint

This viewpoint lists the best available technology currently deployed. If new technology is employed which is not listed then it should be added to the list later. It is very helpful keeping VO consistent with the upcoming demands from business and user requirements and advancement in new computing paradigms and methods. Platforms used for collaboration have remained in a constant up-gradation. Choice must be made on technology by giving weighage to QoS, security, cost effective and timely solution to the end user. An important sub-viewpoint of technological aspect is virtualization. It provides the way to reuse hardware cost, respond dynamically and maximize resource utilization and easy relocation. Virtualization viewpoint deals with logical resources rather than physical resources.

All these viewpoints are shown in the diagram Figure 3.4 These viewpoints can be represented using Lists, Tables, UML tools, and other requirement specification tools available. They are also extendable and organizations can add any further categories according to their goals.

3.7 Interface Description of Components

RAVO is composed of multiple layers and each layer provides a set of components which are the building block of a VO. Selection of these building blocks is subjected to various aspects (i.e. life span, nature (dynamic or static), type, formal, informal and so on). We define interfaces for these components by specifying parameters (mandatory and optional), methods and necessary conditions for their executions.

3.7.1 VO Specifications

VO needs to keep specific information in general, when created. It possesses some characteristics, (e.g. Unique ID, Date Created, Description about purpose domain etc). It also requires to maintain information about participating organizations and individuals. To create a VO, the necessary information to be maintained is detailed in Table 3.1.

Table 3.1: *Virtual Organization: Interface Specification*

Entity Name: Virtual Organization		
Attributes /Modules	Description	Mandatory /Optional
VO_ ID	Unique identity assigned to a VO	M
VO_ Description	Brief description about , purpose, domain etc	M
Date_Created	Date when VO was created	M
Life_Span	Duration for which VO is created	M
Date_Expire	Expected end date	M
Membership_Criteria	Open, Close, Moderated	M
VO_Type	Public VOs are visible to everybody. Membership criteria will be open. While non-public VOs are open to members only. Membership in this case will closed	M
Add_Member()	According to the type of VO, this module adds a member	M
Delete_Member()	Removes a member according to the specified conditions	M
Update_Member()	Updates the existing information of members	M
Participating_Entities _information()	Keeps information about the partner entities which make a VO. Depending upon the context these modules can have variant information, which is left open for the developers. Including their targets, resources, members, contribution and consumption costs agreed upon.	M
Terms_of _Collaboration()	Rules to be agreed upon before becoming a partner/member of a VO	M

3.7.2 Resource Provider Information

It is a must to maintain and update the information about the resource providers in a VO. Organization offering resources, time period for which resources are made available and access rights are potential characteristics. Details are shown in Table 3.2.

Table 3.2: *Resource Provider Infromation*

Entity Name: Resource Provider Information, Mandatory		
Attributes /Modules	Description	Mandatory /Optional
Resource_Provider_ID	Unique ID	M
Organization_Details()	Company or individual details	M
Time_period	Duration for which resources are provided to the VO	M
Access_rights	Defined in Business Model or Contract	M
Usage_details()	Business Model conditions and terms for providing the resources/ Free of cost in case of non-profit organization	M

3.7.3 SaaS Layer

SaaS layer of RAVO consists of optional and mandatory components. Choice of components and decision on their status (mandatory and optional) is open for the developers. The inclusion of components is dependent on the requirement definition by the stakeholders. SaaS Layer has one layer, named Service layer. Its components are defined in the following section separately.

3.7.3.1 Query Interface

RAVO proposes Query Interface as a mandatory component at Service Layer. User is facilitated with remote or desktop access. Query Interface enables user to search for their problem solution in Knowledge base. Knowledge base contains history of problems solved previously. On successful query user is provided with appropriate output. In case of no matching solution, query is processed and problem solutions is provided to user and Knowledge base is updated. Query Interface must provide login facility, identify the query type, check for existing solutions and must maintain a tolerable response time. Details are shown in Table 3.3.

3.7.3.2 Domain Specific Application (DSA)

DSA is a mandatory component DSA provide user with the ability to either download the applications and run on their own systems or use them at VO platform for problem solution. The range of applications depends on the domain and type of VO. Stakeholder can share their applications paid or non-paid basis. Sharing of application can be conditional (e.g. fully or partially paid in case of profitable organization). Information maintained about DSA must include how it is accessible (online or offline), access rights and cost (as defined in the Contract/Business Model). Further details are shown in Table 3.4.

3.7.3.3 Data Mining Tools

Data mining tools are an optional component of RAVO. They are a must for analytical and scientific research based VOs. Interface for data mining tools include tool description , access rights and manul/help. Specifications are given in Table 3.5.

3.7.4 PaaS Layer

PaaS layer is composed of two layers, namely 3-VO Layer and 2-Abstract Layer. Component Specification is detailed below.

3.7.4.1 VO Trust

VO Layer consists of two main mandatory components. VO Trust is the most important of all components. It is formed by combining different modules and performs multiple tasks. It is responsible for *Authentication* and *Authorization* of VO members. Authorization is done on the basis of *Roles* defined in the *Contract/Business Model*. VO Trust have a mandatory emphContract which consists of policies to achieve the goals of VO. In profitable or partially profitable VOs Business Model is also mandatory component of VO trust. In RAVO Business Model is optional and depends on the type of VO, however Contract is mandatory. Access rights are defined in contract or Business Model. Different methods are available to define the access rights. Organization models and access rights are comprehended in [98]. According to the authors access rights might be subjected to *Organizational* and *Direct* change [98]. All components of VO Trust are synchronized to maintain the VO. Each component is assigned a specific task and output of one component provides input to the other component. VO Trust has a *Resource Information* component that acts like a *Registry*. It keeps necessary details about all the resources available in VO. These components are shown in Table 3.6

3.7.4.2 User Interface

User Interface is a mandatory component of VO Layer. It provides access to the platform services offered by VO. User is authenticated and authorized using Login option. After authorization, user can formulate different queries and perform actions. These facilities are realized using a Web portal. Details are shown in Table 3.7.

3.7.4.3 Workflow Tools

Abstract Layer is a sub layer of PaaS layer. It includes different components. Workflow tools is a mandatory component of this layer. Workflow management is a critical aspect of a VO in any domain. It supports Provenance management which plays vital role in monitoring and maintaining a VO. Workflow can be interpreted in different forms (e.g. graphical, textual, source code). Interpretation mode is chosen on the level of audience a VO possess. Workflow tools keep track of all the processes active in VO. Process management can be included as a sub component of a Workflow Tools. Dynamic adaption of in-process workflow is an essential part of any workflow management system. Classification of approaches along their strength and limitations used for dynamic adaption in workflow systems are detailed in [99]. Flexibility criteria in process management to handle the foreseen and unforeseen behaviors are categorized in [100].

Workflow tools allow user to define workflows for a problem solving activity. The participants responsible at each stage of this activity are notified and are responsible for delivering the promised results. Workflows are reusable and reduce redundancy and time for similar problems. Information maintained consist of workflow ID, type, status, access rights, how it interprets the results and process management. workflows are used by Provenance management to track the problem solving activity on user demand. Details are shown in Table 3.8.

3.7.4.4 Provenance Tools

With the advent of financial computing systems, as well as of data-intensive scientific collaborations, the source of data items, and the computations performed during the incident processing workflows have gained increasing importance [101]. Provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource[9]. In a VO, provenance forms a critical foundation for enabling trust, reproduction and autentication. Provenance assertions are a form of contextual metadata and can themselves become important records with their own provenance[10]. Provenance Tools are mandatory and included in Abstract Layer of RAVO. Provenance management is dependent on authorization, query management and workflow management which are listed in Table 3.9.

3.7.4.5 Graphical Interface

Graphical Interface is a mandatory component of Abstract Layer. It facilitates users to perform different task in VO Web portal. It provides an understandable interface to interact with the VO Details are shown in the Table 3.10.

[9] http://www.w3.org/2005/Incubator/prov/wiki/
[10] http://www.w3.org/2005/Incubator/prov/wiki/

3.7.4.6 Resource Management

Resource Management is a mandatory component of Abstract Layer. It provides a mechanism to select and aggregate resources for a problem solving activity. Depending upon the underlying technology, VO developers can deploy different resource management tools. Necessary information maintained depends on the resource type and interest of participating entities. Basic information includes resource's unique identification, categorization as logical or physical, owner information, access rights and costs etc. RAVO being technology independent lists a brief description in the Table 3.11.

3.7.5 IaaS Layer

IaaS layer is composed of Infrastructure Enabler Layer and Factory Layer. This layer from the fabric of RAVO. All the resources are avaialble in Factory Layer and are exploited through Infrastructure Enabler Layer.

3.7.5.1 Infrastructure Enabler

This module is depending on the underlying technology . QoS, Service Level Agreement (SLA) , Security, Fault tolerance and Disaster management are most important issues. These aspects have to be implemented on the bases of terms and conditions presented by participating entities. Financial aspect is another limitation for the implementation of these modules. Any other desired aspects can be added to extend the Infrastructure enabler layer. Components shown in Table 3.12 are dependent on the decision of the developers. RAVO identifies least basic and gives developers an open end to use them as mandatory or optional in their target VO.

3.7.5.2 Resource Catalogue

This module is part of Factory Layer but not explicitly shown in RAVO. It acts like a database for the resources. VO developers can include it at any layer according to their needs. RAVO keeps it at the Factory Layer as a mandatory component. It contains information about resource management and Table 3.13 presents it in detail.

3.7.5.3 Expert

Expert represents the logical resource in RAVO Factory Layer. An Expert plays important role in problem solving activity. Expert can be contacted online during the problem solving process or she can be accessed offline. VO must maintain detailed information about Expert so that this feature can be fully exploited. Details are shown in Table 3.14.

3.7.5.4 Data Service

Data Services is a mandatory component of Factory Layer. It represents the physical resource in RAVO. Data stores are important scientific and research based VOs. Details of this components are available in Table 3.15.

3.7.5.5 Computational Services

Computational Services are mandatory component of Factory Layer. They also form the physical resources offered by a VO. Details are specified in Table 3.16.

3.8 Summary of Research Contribution

This chapter elaborated our proposed RAVO. An overview of the existing efforts in this specific area were listed and compared with RAVO. Our proposed work is distributed as follows: Requirement analysis phase, Component identification phase, Generic framework, Viewpoints and interface specification of components. In requirement analysis phase, VO developers start their quest. First part consists of two phases. Phase I provides a series of critical questions, which defines purpose and justifies the goals. Phase II takes developers one step ahead by identifying the main components of a VO. Identification of mandatory and optional components is done in Interface specification section.

Second phase is providing a framework to integrate these concepts. RAVO reduces time and effort for building a VO by identifying the steps in an understandable manner. Viewpoints are detailed and Interface specifications are presented in a simple tabular form. Another important aspect of RAVO is to let developers choose the underlying technology which suites to their organizational and financial limitations. Next chapter details the Stakeholder and Generic View of Resources in the context of VO.

Table 3.3: *Query Interface*

Entity Name: Query Interface, Mandatory, 4-Service Layer		
Attributes /Modules	Description	Mandatory /Optional
Access_Mode()	Desktop, Mobile	M
Login()	Login to authenticate members of VO	O
Query_Processing()	Responsible for activities from query initiation to solution output	M
Query_Type()	Categorize according to the resource offered. An online expert opinion, download, resource request	M
Existing_Solution()	Searches the knowledge base of VO for existing solutions on the basis of parameter provided in the Query type. Successful search is return a problem solution. Unsuccessful search branches control to the VO management for finding a new solution from the scratch.	O
New_Solution()	It finds solution of the proposed problem (if Existing_Solution() is unsuccessful). User is provided with the appropriate output according the query	M
Response_Time()	Urgent/Normal, the user must be provided with a time frame depending upon the query type.	M
Input_Data()	Query string, necessary parameters	M
Output_Result()	Give back results to user. It could be notification as an email, a document, or a link to the Web site where results can be found. Resource access permission, unsuccessful processing status, contact information of an expert, Or any other method agreed upon by the participating entities	M

Table 3.4: *Domain Specific Application*

Entity Name: Domain Specific Application, Mandatory, 4-Service Layer		
Attributes /Modules	Description	Mandatory /Optional
Application_ID	Unique Application ID	M
Type	Standalone software, Online executable only,	M
Access_rights	Who can access this application	M
Application_Details()	Name, version, owner, volume, PC/mobile application, compatibility (OS support, memory etc)	M
Status	Free ware, trial, open source, paid	M

Table 3.5: *Data Mining Tools*

Entity Name:Data Mining Tools, Optional, 4-Service Layer		
Attributes /Modules	Description	Mandatory /Optional
Tool_ID	Unique Tool ID	M
Purpose	Details of how this tool works and for what purpose	M
Access_rights	Who can access this application	M
Tool_Details()	Name, Version, Owner,	M
Manul()	A guide or instruction set for user explaining how it can be used efficiently	M

Table 3.6: *VO Trust*

VO Trust, Mandatory, 3-VO Layer		
Attributes /Modules	Description	Mandatory /Optional
Authentication()	Authenticates user as a VO member	M
Authorization()	Verifies the access rights assigned to the member according to a given role	M
Contract()	Contains sub modules i.e. Policy(), Goal(), Role(), Feedback()	M
Business_Model()	Contains sub modules Rules(), Roles(), Pricing_Algorithm(), Goal()	O
User_FeedBack()	Feed back from stakeholder is utilized to enhance the contract or Business Model. Change In requirement must be incorporated in contract or Business Model to keep the VO updated and evolve them dynamically	M
Resource_Infromation()	Resource Management() and Resource Catalogue()	M

Table 3.7: *User Interface*

Entity Name: User Interface, Mandatory, 3-VO Layer		
Attributes /Modules	Description	Mandatory /Optional
Login()	Authentication and Authorization	M
Query_Management()	Taking input parameters, processing query, displaying results processing query, displaying results taking feedback from user	M

Table 3.8: *Workflow Tools*

Entity Name: Workflow Mandatory, 2- Abstract Layer		
Attributes /Modules	Description	Mandatory /Optional
ID	Unique Workflow ID	M
Description	Sequential, state machine, data driven	M
Status	Start, end, proceeding, paused	M
Authorization_Information()	Who have right to access and call this module/Association with Roles	M
Interpretation_of_Workflow()	How Workflow provides information to the stakeholder /graphical, textual, source code, depending upon the mode it contacts other modules in the workflow management system to represent the information in an understandable form (code, markup languages, or a combination of both code and markup to author workflows.) Choice of approach depends on the authoring mode requirements for the solution.	M
Process_Management()	Includes Instance_Management() that controls the individual process instances to manage the concurrency	M

Table 3.9: *Provenance Tools*

Entity Name: Provenance, Mandatory, 2- Abstract Layer		
Attributes /Modules	Description	Mandatory /Optional
Provenace_ID	Unique identification of the module	M
Provenance_Management() M		
Authorized_Access()	The user requesting for Provenance regarding a problem is authorized and have access rights. On successful authentication request is processed accordingly	M
Query_Input()	Parameters required to execute provenance	M
Query_Type()	Type of request made, what type of provenance is needed	O
Query_Processing()	Requesting related modules for processing Query (e.g. Workflow_Managemenet(), Resource_Management(), etc) depending upon the nature of query, modules are contacted	M
Workflow_Management()	Defined in workflow entity	M
Output_Results()	Sends the results in the desired format. Graphical user interface is used to assist the provenance mechanism, where user can formulate the query and processing details are hidden and results are displayed in a user understandable format (flowcharts, graphs etc). Complex formats must be available of request(code or encrypted languages). Incase of unsuccessful processing, proper messages must be conveyed to user.	M
Update_Provenance database()	Solutions are stored in the provenance database for future reference and to reduce the time for similar problems.	M

Table 3.10: *Graphical User Interface*

Entity Name: Graphical User Interface, Mandatory, 2-Abstract Layer		
Attributes /Modules	Description	Mandatory /Optional
GUI_ID	Unique ID	M
Input_Management()	Controls the input parameters for user interaction	M
Processing_Management()	Controls the details (parameters) flowing among different modules	M
Output_Management()	Controls how results are displayed to the user and stored for the future use	M

Table 3.11: *Resource Management*

Entity Name: Resource Management, Mandatory, 2- Abstract Layer		
Attributes /Modules	Description	Mandatory /Optional
Resource_ID	Unique resource identification	M
Resource_Type	Logical/Physical	M
Availability_Status	Resource is active part of VO	M
Resource_Provider _Information()	Resource provider information is maintained	M
Resource_Cost()	Resource usage policies or Business Model, which maintains resource cost and usage. Free in case of non-profit VO	M
Access_Rights()	Defined in Contract/Business Model	M
Resource_Scheduling()	How resources are aggregated for a problem solving activity.Different methods and algorithms are developed for this purpose	M
Resource_Consumption()	Percentage of the resources consumed in a problem solving activity	O
Resource_History()	Early participation in a problem solving activity and performance	O
Resource_Maintainance()	Add_resource(), Update_Resource(), Remove_Resource()	M

Table 3.12: *Infrastructure Enabler Layer*

Entity Name: Infrastructure_Enabler, Mandatory, 1-Infrastructure Enabler Layer		
Attributes /Modules	Description	Mandatory /Optional
QoS_Management()	Manages Quality of Service parameters agreed upon by participating organizations	M
SLA_Management()	Manages SLA agreed upon by participating organizations in Business Model or Contract	M
Security_Management()	Provides Security mechanism, secure communication and encryption facilities	M
Fault_Tolerance _Management()	Manages fault tolerance and disaster management, how to degrade gracefully instead of being crashed	M

Table 3.13: *Resource Catalogue*

Entity Name: Resource Catalogue, Mandatory, 0-Factory Layer		
Attributes /Modules	Description	Mandatory /Optional
Category_ID	Identifies the category to which a resource belongs	M
Category_Type	hardware, Software, Logical	M
Status	Available or Not available	M
Resource_ID	Composite ID : Category_ID and Resource_ID	M
Resource_Type()	Computational, Storage, Data, Expert, Multimedia (Document, Audio, Video etc)	M
Access_Rights()	Defined according to the roles defined in Contract/Business Model	M
Add_Resource()	Resource Management	M
Remove_Resource()	Resource Management	M
Update_Resource()	Resource Management	M
Resource_Provider _Information()	Detailed information about the resource provider. Accessed via Resource_Provider_ID	M
Usage_Policy()	Details usage details and calculates cost for resource consumption. Legal terms and conditions associated with Resource. Resource provider also maintain these details for record.	M

Table 3.14: *Expert*

Entity Name:Expert Mandatory 0-Factory Layer		
Attributes /Modules	Description	Mandatory /Optional
Category_ID	Identifies the category to which a resource belongs	M
Category_Type	Logical	M
Resource_ID	Unique resource ID	M
Expert_Profile()	Details about expertise, domain, association/affiliations	M
Contact()	Email, Phone, Fax, timings of availability for online assistance	M
Availability_Status	Online/offline	M
affiliation	Individual or with en enterprize	M
Role_Assigned()	Stakeholder role (Subject/consumer/producer/administrator)	M
Resource_Provider_ID	In case of expert belonging to a participating organization	M

Table 3.15: *Data Services*

Entity Name: Data Service, Mandatory, 0-Factory Layer		
Attributes /Modules	Description	Mandatory /Optional
Category_ID	Identifies the category to which a resource belongs	M
Category_Type	Physical	M
Resource_ID	Unique resource ID	M
Availability_Status	Up/Down (resource is working correctly or not)	M
Resource_Cost()	Usage cost of the Data service	M
Access_Rights()	Authorization for utilizing Data service according to the Role assigned	M
Resource_Provider_ID	Unique ID	M

Table 3.16: *Computational Services*

Entity Name: Computational Services, Mandatory, 0-Factory Layer		
Attributes /Modules	Description	Mandatory /Optional
Category_ID	Identifies the category to which a resource belongs	M
Category_Type	Physical	M
Resource_ID	Unique resource ID	M
Resource_Provider_ID	Unique ID	M

4 Stakeholder and Generic View of Resource in Virtual Organizations

4.1 The Resource Hierarchy

The existence of a VO is typically identifiable by many individuals, ad-hoc groups, research teams, and national and international organizations deploying a wide range of resources [3]. Initially, resources were meant to be hardware such as storage, high performance devices (measuring earth quake, weather forecast, printers, etc), and software (applications, utilities, simulation facilities) [10]. The extensive use of computer technology for problem solving changed the nature of resources [102]. Now resources are distributed as *logical* and *physical* resources. Defining a resource in a VO environment is dependent on the participating entities and domains in which the VO operates. A categorization of resources is presented in Figure 4.1 [5].

In our research endeavor, a complex but interesting relationship was discovered between user roles and resources [1]. During resource consumption and contribution, at a certain point, user roles and resources are interchangeable. Some may find these concepts overlapping. Previously, resources are purely considered to be something, which is being consumed by the user, as shown in Equation 4.1.

$$USER \overset{Consumes}{\leftarrow} RESOURCE \tag{4.1}$$

However, resources are also contributed by users in a problem solving activity. This situation is defined by Equation 4.2.

$$USER \overset{Contributes}{\rightarrow} RESOURCE \tag{4.2}$$

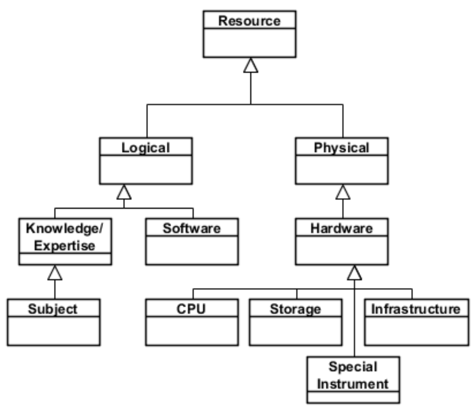

Figure 4.1: *Resource Hierarchy in Virtual Organization*

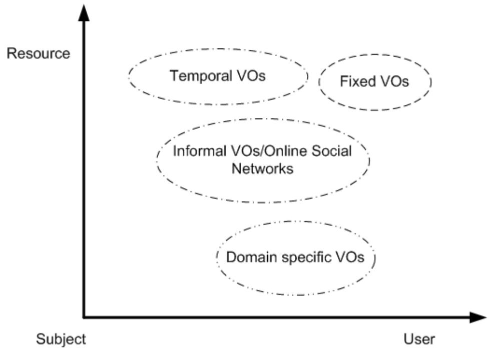

Figure 4.2: *Subject, Resource and User in different VOs*

The situation becomes even more complex, when a resource itself is a user. For example, in an exploration scenario, a meteorologist wants to know the reason that causes an unexpected storm. By chance she is the member of LEAD VO [54] [95]. So she searches for available data sources, and utilizes the tools offered by this VO for the analysis. In case of non-satisfactory results, she consults an expert for guidance and performs an analytical activity with changed data sets. In this scenario, the expert opinion is used as a resource, while experts also utilize VO resources for gaining knowledge [95]. From a non-scientific perspective (e.g. online social networks), the same situation can be easily identified in group discussions. A member asks a question and other members share their experiences, which can provide potential solutions to the problem, and vice versa. So the equation takes the shape as shown below

$$RESOURCE/USER \rightleftharpoons RESOURCE/USER \tag{4.3}$$

Even more according to our definition of Subject, the equations above can be generalized to Equation 4.4

$$SUBJECT \Longleftrightarrow SUBJECT \tag{4.4}$$

Here, the user is consuming the knowledge of an expert, who acts both as user and resource. Subject is the notion given to a user who itself can be used as resource. There are two reasons for choosing this term. First, a Subject (user) initiates an activity in the VO environment and secondly, a Subject (resource) is under consideration to be useful in a problem solving activity. Figure 4.2 shows the Subject, resource and user relationship in different types of VOs.

4.1.1 Stakeholder

Viewed differently, an organization consists of different elements that work together for common goals. An element of an organization can be "physical", if it is visible and appears in the organizational structure of the organization. An example of a physical element is a person or an office of an organization (physical boundary). An element can also be "virtual" or "unseen" such as knowledge or expertise, financial asset flows or information. These elements are virtual in the sense that they are not necessarily visible, and may not appear in the organizational structure but play the role of a “glue” in organizations. In fact, in an organization, a nonphysical element is something utilized by the organization and represents an aspect with which the physical elements can complement each other. Moreover, these physical elements are generally connected or linked by one or more virtual elements in order to create a unity. For example, in a paper co-authoring organization, in order to produce joint articles, different authors residing in different places may work “together” as they possess certain complementary knowledge [22]. Stakeholders are an essential part of the VO. Resources offered by a VO are consumed and produced by stakeholders. Any change in the requirements from designated community can change the structure of VO. Definition of stakeholder in VO environment is domain dependant. The IEEE Standard 1471-2000 [62] defines the stakeholder as

- The user of the system.
- Those responsible by the acquisition of the system.
- The developers and providers of the system’s technology.
- The maintainers of the system as a technical operational entity.

According to this description, stakeholders are classified into following four categories: consumer, contributor, developer, and administrator in a VO. Resources offered by VOs are utilized by stakeholders in these four capacities. VOs offer globally distributed resources to its users. With the technological shift, resources offered by a VO are also changing. The relation between user and resource is partially overlapping. This situation motivated us to review the user’s roles and resources offered in a VO. We introduced a new term for a special type of resource in VO in [1] [2] [3], which we called *Subject*. We use the Unified Model Language (UML) approach to detail these concepts.

Figure 4.3 explains the stakeholders who constitute a VO. Stakeholder class is divided in to two broad categories Enabler and Subject.

- Enabler: This class represents the technical part of stakeholder in VO. It represents the group of people who are necessary to maintain the environment and update tools, software and hardware time to time. They enable the resource sharing mechanism to run smoothly. This class is further divided into three subclasses which are detailed below:
  - Developer: The Developer class includes the professionals and application developers from participating organizations. There can be professionals who contribute open source software to the improvement of IT support in specific domains. However, any person can contribute knowledge in form of applications in a specific domain, even if they are not member of a participant organization.
  - Administrator: Another potential subclass is Administrator, who monitors the VO platform for smooth use and in case detects and manages hardware

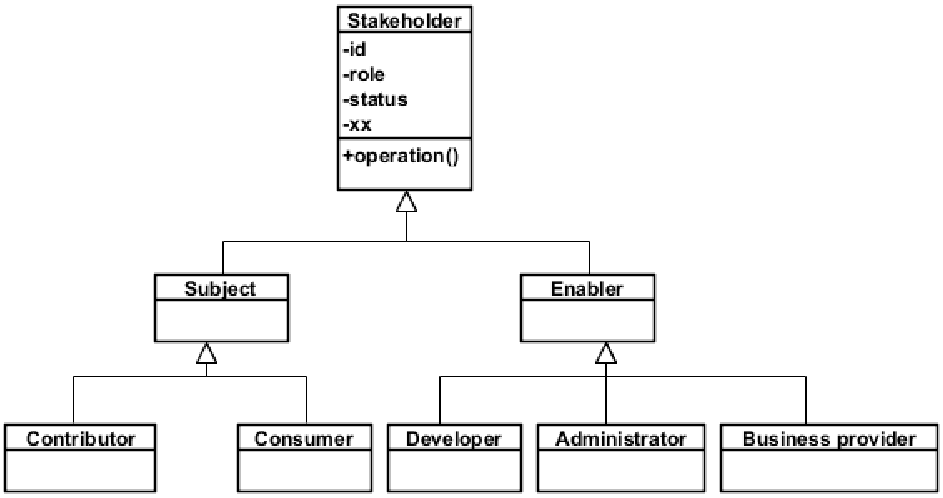

Figure 4.3: *Stakeholders in a Virtual Organization*

and/or software crashes. Again the Administrator class can be represented by a group of paid experts, who are specialized in their respective jobs assigned. Participating organizations can hire such professionals to monitor the services they are offering to the VO.

- Business Provider: This class is optional depending upon the context of VO. In E-commerce, economics and business collaboration, this subclass is a must. VO has a Business Model and cost of resource utilization and access is defined in accordance with participating entities. Business provider makes it possible to rent resources from third party and provide it to the customers. In education, entities from industry can provide their resources for research which is paid. Business provider can negotiate to make the delivery of tools, software and hardware possible to the project teams and researchers.

- Subject: VOs offer globally distributed resources to its users. With the technological shift, the resource type offered by a VO are also changing. The relation between user and resource is partially overlapping. This situation motivated us to review the user's roles and resources offered in a VO. We introduced a new term for a special type of resource, called Subject in VO in [1] [3] [2]. Subject is defined as "a component of VO, which can consume the resources, offered and also can act like a resource to be consumed in the VO environment". It contributes and consumes resources and itself can be viewed as a resource of the system. Therefore, a Subject resembles a generic block of a VO which results into a new definition for VO. Thus we propose VO as "a set of cooperating building blocks, called Subjects".

Subject has two subclasses, Consumer and Contributor:

- Consumer class represents the users who utilize the resources.

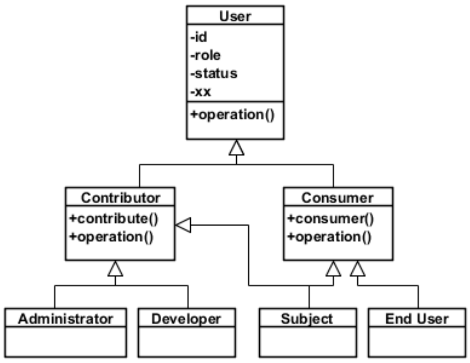

Figure 4.4: *User Roles and their Dependencies*

- Contributor class is the collection of users who are adding resources to the VO. This approach places the user at the center of the design and whole framework evolves as the user role and requirements change.

4.2 Subject as a Resource

The previous section established the human expertise as a resource in a VO. Now we will present a template for a Subject class with its functions and relations to other user types in a VO. User types are a must for the system, because it helps in many ways, e.g., by defining trust, building a Business Model/Negotiation Model, setting security, authorization at different levels, managing the incoming and outgoing traffic (in dynamic environments), consumption and contribution of resources, and many more. In VOs a user is given a role according to a Business Model or negotiation pattern for collaboration. Roles may vary as the target domain changes. There are few characteristics and activities that can be generalized. Figure 4.4 presents the classification model of a *user* in a VO environment. This model covers both formal and informal types of VOs. A class diagram is created using Unified Modeling Language (UML) 2.0 to present the pattern.

A User is the superclass with attributes defined in the context such as

- Id: string (any combination used for authentication)
- Role: string (assigned role in the said VO)
- Status: string (active, passive)

A User class is further specialized into two broad categories of **Contributor** and **Consumer**.

- The **Contributor** class presents the instances of a user, who contributes to the VO. The main method is **contribute()**. This class is further divided into **Developer** and **Administrator**, performing pure contribution and no utilization. Here, the **Subject** is also a subclass of **Contributor**. It realizes the role of a contributor while it can also act as a consumer in a VO environment. It inherits the attributes of the main **User** class and provides contributing methods (functions). The type of contribution can be added according to the role of the user and the domain in which it is active.

  - **Administrator:** Another potential subclass is Administrator, who monitors the VO platform for smooth use and in case detects and manages hardware and/or software crashes. Again the Administrator class can be represented by a group of paid experts, who are specialized in their respective jobs assigned. Participating organizations can hire such professionals to monitor the services they are offering to the VO.
  - The **Developer** class includes the professionals and application developers from participating organizations. There can be professionals who contribute Open Source Software to the improvement of IT support in specific domains. However, any person can contribute knowledge in form of applications in a specific domain, even if they are not member of a participant organization.

- The **Consumer** class represents the class of users, who just utilize the resources by performing pure consumption only. This class has two subclasses called **End User** and **Subject**. It contains a method **consume()**, which shows that an instance of this class will be able to consume the resources offered by the VO.
  - The **End User** class represents a set of users who only consume the resources provided by the VO.
  - The **Subject** class represents the category of users who utilize the resources and also contribute to the VO environment. Currently two generalized methods are associated with the Subject class namely, **contribute()** and **consume()**. An instance of the Subject class is capable both to utilize the resources offered and to contribute to the VO at the same time. Instances of the Subject class can act as a Consumer or Contributor (as a resource), who share partial characteristics of their superclass.

A Business Model can also be developed on the basis of this categorization. Users belonging to the **Subject** class, can be given a high priority. This priority can entitle them to benefits such as money, free memberships to different participating organizations, utilizations of resources (test beds access, and access to reference material, etc).

The Contributors can have the 2nd highest priority, because they are the paid members of the VO. They develop tools for the maintenance of VO and monitor it. Such users are employed by the system. A possible subcategory could be developers contributing open source applications for the improvement of IT environment. They can be given priorities according to their contribution to the system, e.g., free resource consumption.

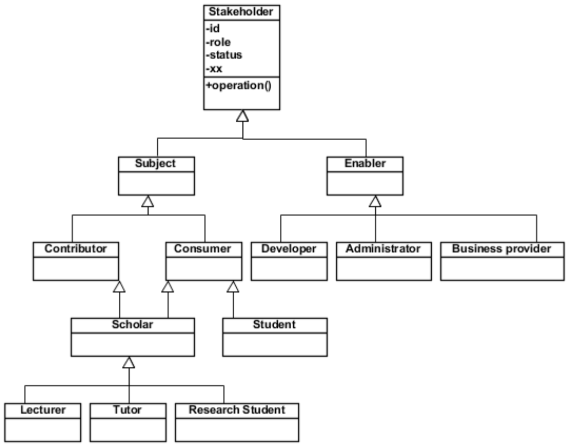

Figure 4.5: *Stakeholder of a Virtual Organization in E-learning System*

4.3 Stakeholder in Virtual Organization based E-learning system

In VO based E-learning environment, we define *Contributors* and *Consumer* as *Scholar* and *Student*. Scholar is further divided as Lecturer, Tutor and Research Student. Enabler category is the essential part and has already been defined in the section 4.2. Figure 4.5 extends the general pattern of stakeholders developed for the RA for VO in the context of E-learning. Actual instances of *Subject* are visible here. Extended categories are detailed below:

- Scholar: This class represents those users who both consumer and contribute to the E-learning system. It is further divided into three other classes namely Lecturer, Tutor and Research Student.
  - Lecturer is responsible to deliver the materials, which are used by the tutors for guiding students. In some situations, a lecturer guides students as she has planned the course. The lecturer also provides course description, setting goals both on teacher's and student's part, preparing lecture material and delivering in classroom. Lecturer is given higher access rights to the resources offered by E-learning environment. Lecturer consumes the existing resources to enhance her skills and contributes not only in the class room, but also share knowledge and guidance thereby acting as a resource itself.
  - Tutor is responsible for following actions.
  - Informs students about the course updates.
  - Arranges meetings for problem solution and discussions both online and offline using email, mailing lists, messenger or even social networks.
  - Provides technical support during lecture (multimedia, stationary, audio/visual support).
  - Books lecture halls. Lecturer and tutor both are in contact to deliver the course contents as efficiently and effectively as possible. Both contribute to the resources in E-learning environment and also consume the tools to enhance their capabilities.

- Research student: Access the E-learning system for material in specific domain. They are given membership in order to provide access to the online resources. They contribute to the system by developing new tools, upgrading the existing applications and research contributions to the E-learning environment. They are given more rights than the students who are at graduate or under-graduate levels. They consume and contribute to the environment and act like Subject. Also, this categorization is the realization of the promised goal presented in future direction in [1]. This pattern supports the idea of a VO for research students and defines their role in the environment. Research students are example of Subject. They consume and contribute to the environment and are viewed as a resource to the system by their research and development in the said domain.

- Student: It belongs to the consumer class. In the context of E-learning, it represents the graduate and under-graduate level student. They use system for utilizing resources only. E-Learning systems provide a variety of facilities; not only course contents (e.g., lectures, videos, e-books, licensed software) but also scheduling tools, uploading assignments, managing their workspace. Registered students are allowed to access these facilities. Students are provided with the pool of resources from their university or department in which they are enrolled. Some groupware technologies are also used to collaborate in the absences of an E-learning environment on part of the institution. Students are asked to join mailing groups to stay connected to the teacher for the updates of a specific course which makes it difficult to keep track for all the courses updates. Proposing a VO based E-learning system benefits both teachers and students to communicate at the same platform in a consistent manner.

4.4 Informal Virtual Organization and Subject

Informal VOs are part of our lives in the form of social networks (e.g., Facebook, Myspace, MyExperiment) [10]. These user-driven networks are typical examples of informal VOs where every user has its own goal for consumption and contribution to the resources pool. Today, online social networks are becoming essential part of everyday life of humans, who have access to Internet. People find it easier to connect to each other using social networks. These social networks can be visualized as a collection of *small scale informal virtual organizations*. Each user is given the right to access a number of resources offered by an online social network by creating a profile. These platforms give a sense of authority to the members by allowing them to initiate different activities. On the other hand, members can participate in the activities initiated by other members. Online social networks are an interesting area to study roles played by members. This paper identifies the resources available in a VO. It reveals the role of users as a resource in an online social network. In the context of this paper, online social networks are presented as a special case of VOs.

User classification provided in previous section can be observed in different domains. We presented the elaborated roles in VO for E-learning [2] and CI [3]. Both are examples of formal VO. *Ambient intelligence* is taking social networking to a new level of awareness [103]. This awareness is propagated from relatively constant contact with one's friends and colleagues via social networking platforms on the Internet. Informal VOs realize the concept of ambient awareness. Social network fall in the category of informal VO. Informal VO are characterized by absence of a specific goal, rather they are user driven [10].

Online social networks are user driven, with no specific goals. However, they can be joined to meet several goals (e.g., making friends, playing games, joining research, religious, social, health, sports groups, to communicate with distant relatives or friends, promote different causes, advertise, participate in discussion forums etc). Goals can be anything supported by the platform. Here, it can be clearly observed that every user is a resource of this informal VO. It exists only due to the relationship between the users and improves with the feedback they provide. Popular social networking websites are Facebook [104], Myspace [105], Twitter [106], and Blogger [107], etc.

To justify the patterns developed in the previous section, we choose Facebook as an example of an informal VO. Facebook [104] is a popular online social network launched in February 2004. It is selected as an example to identify the roles and resource dependencies in informal VOs. The activities performed by users are:

- Create a profile, update and set privacy settings, delete and add applications.
- Add people as friends (send, reject and accept requests).
- Send and receive private and public message.
- Notify of updating to friends.
- Define status settings.
- Chat with online friends.
- Make lists depending upon privacy settings.
- Add photos.
- Add videos.
- Create notes.
- Join networks organized by workplace, school, or college.
- Like fan pages.
- Join and start groups, networks.
- Send a virtual "poke" to each other (a notification in turn tells a user that they have been poked).
- Send gifts.

- Visit marketplace.
- Play games.

In social networks every need or goal is dependent on another user. If a user wants friends, so this user is looking for a resource (friend). She plays games, which are provided by other users (in most cases). She joins a cause, which is initiated by another user. In any of the above listed actions a user needs other users and their expertise or shared information to fulfill her needs.

On the other hand, information, expertise, material, pictures, videos provided by her can act as a resource for other users. She can initiate fan clubs, discussion groups and any cause, to invite people and grow her community. A use case was developed to understand the activities performed in this informal VO, shown in Figure 4.6. The user roles and their interaction with existing resources is detailed below.

- **Subject:** The role of a user as a resource is more profound in an informal VO than in formal ones. This is illustrated clearly in our current example of Facebook. A user creates an id and is given right to perform several activates, as listed above. Here the user is a contributor and a consumer herself. For example, a member uploads a video or photo or creates a note, which is being watched by other users and vice versa. Sending and receiving friend requests, messages (open and private), initiating groups, causes and campaigns, joining groups, reading and writing notes, sending and receiving gifts, communicating with friends through wall, and chat and status updates are the activities performed as *Subject*.

  A Subject also gains information from news feeds. An interesting facet are business promotions, which play the role of End user. Many products are introduced to E-communities using social networks by their manufacturers. Facebook is also used by different manufacturers to reach their customers. News channels, media, health, education, research communities, etc., all use social networks according to their requirements and goals.

- **Developer:** Members also play games, utilize applications developed and contributed by developers to the platform.
- **Administrator:** Group of specialized person(s) maintains the platform for performance, backup and routine maintenance.

4.5 Summary of Research Contribution

This chapter presented the concept of resources and users in both formal and informal VOs. A resource hierarchy is defined and the role of a user as a resource was observed and discussed in different environments. The understanding of user roles is necessary for building a trust model for VOs. This approach was extended by a generic pattern for users in VOs and was justified using online social networks (e.g., Facebook). The concepts are elaborated with examples to understand when a

Figure 4.6: *Facebook: User Roles*

user changes her role from a consumer to a resource and starts contributing to the environment. Hence the term *Subject* was justified.

Online social networks provide resources to its members. Every member contributes to the community silently. The impression of a member as a consumer is fading by growing needs of "give and take" collaborations. This new concept of *Subject* fits well into the nature of online social communities. It will help in the future research on VOs to understand the concept of a Subject as a fix-point where users and resources become the same. It will also set the bases of user roles in designing a RAVO as our future direction. The next chapter details application of RAVO in different comains. Concepts of stakeholders and *Subject* are verified in E-learning, CI and CS.

5 Application Domains

RAVO is presented as an outcome of theoretical and practical implementation of the concepts. We developed concrete models in the domain of E-learning and Computational Intelligence (CI). Refinement of concept was applied to informal VO and *Subject* was justified in various domains. This chapter provides a brief introduction of the chosen domains, existing system in the respective domains. It also elaborates the use cases developed in the context of domain specific VO by building the concrete models.

5.1 Candidate Systems

We present the existing systems which were selected to instantiate RAVO in different domains.

5.1.1 N2Grid

N2Grid [94] [108] is a system for the usage of NN resources on a world-wide basis. The approach employs the infrastructure of the grid as a transparent environment to allow users the exchange of information (NN resources, as NN objects and NN paradigms) and exploit the available computing resources for NN specific tasks leading to a grid based, world-wide distributed, NN knowledge and simulation system. The system aims to implement a highly sophisticated connectionist problem solution environment within a *Knowledge Grid* and uses moreover only standard protocols and the available technology of so called Web Service to provide a wide dissemination of this grid application. Thus the N2Grid system is, simply speaking, an artificial NN simulator using the grid infrastructure as deploying and running environment. It is an evolution of the existing NeuroWeb [109] and NeuroAccess [110] systems. The idea of these systems was, to see all components of an artificial NN as data objects in a database. We extended this approach by identifying them as resources of the world-wide grid infrastructure. Accordingly to the definition of the notion of "information" of Gundry, we developed a layered grid architecture based on the dimensionality of information in focus which allows differentiating three different grid layers:

- Data Grid, 0-dimensional. The Data grid builds the basis layer and stores data which represent just facts.
- Information Grid, 1-dimensional. The Information grid collects data of the Data grid in a structured manner and attributes it with semantic contents.

- Knowledge Grid, 2-dimensional. The Knowledge grid provides problem solution mechanisms on the administered information allowing a human for acting, deciding or planning.

In this architecture, each layer (starting from the data layer) provides its functionality to the next layer in form of specific services. N2Grid is based on service oriented architecture and spans all three layers of the grid layer architecture.

5.1.1.1 N2Cloud

N2Grid system is chosen as a candidate for applying our proposed approach. N2Grid has been migrated to cloud by a parallel research effort at University of Vienna. We brief the extension here. N2Cloud [96], a novel cloud-based NN simulation system, which provides and exchanges NN knowledge and simulation resources to and between arbitrary users on a world-wide basis following the Web 2.0 principle. N2Cloud enables the exchange of knowledge, as NN objects and paradigms, by a VO environment and delivers ample resources by exploiting the cloud computing principle. The system provides a transparent environment to allow even naive users to exploit the resources of this simulation system. N2Cloud uses standard protocols and is based on a pure Service Oriented approach. Hereby it integrates into the up-to-date service stack (SaaS, PaaS, and IaaS) of service oriented architectures [96]. N2Cloud, a cloud-based application that will enable the CI community to share and exchange the NN resources enabled by the cloud computing paradigm. We gave an overview of the application by highlighting the interaction between the N2Cloud components. To the best of our knowledge N2Cloud is the first cloud-based scientific application in the CI community enabling this new computing paradigm. N2Cloud is a prototype having quite some room for further enhancement [96].

5.1.2 Cooperative Environment Web Services (CEWebS)

Our ideas of supporting VOs in the CI manifest focus around a modular system developed at Institute of Knowledge and Business Engineering of the University of Vienna, called CEWebS [111]. CEWebS[1] stands for Cooperative Environment Web Services and realizes a distributed architecture that facilitates short development cycles and the ability to move new functionality to courses as well as research groups very quickly.

5.1.2.1 Idea and Goals

The CEWebS idea is based on the following assumptions:

- Big organizations (e.g. the University of Vienna) support a multitude of platforms and tools that cannot be unified.
- For the successful adoption of VOs it is necessary to create tools that are specifically designed for special purposes. E.g. the high degree of formalization in

[1] http://www.cewebs.org/

the natural sciences and the computer science easily allows for the creation of tools that can facilitate interaction. To really have an impact, these tools need to extend the common elaborate/upload/review schema.

- Tools should not only be data sinks but should interact with the users or facilitate interaction between the users.

The multitude of special fields (or departments in the university context) leads to a multitude of requirements, which cannot be met by a single system. At the University of Vienna at the organizational level, there are currently two main E-learning systems supported: Blackboard Learning System[2] and Moodle[3]. There are no guarantees that they are not replaced by any other systems in the future. Al- though these E-learning systems cover a similar basic set of tools, additional requirements lead to a need for constant adaption and extension. Therefore the situation occurs that often local system solutions are created. These solutions tend to have the following properties:

- The knowledge that is inherent to these solutions makes them the optimal instruments for teaching. Everything else is measured by the degree of automation and the quality of teaching achieved with these tools. Every extension and/or improvement of the embedding learning platform is refused, if the local solution is affected ("never change a running system") [111].

- Sharing a solution with others is difficult, because normally it blends not very well with different platforms or tools. So a conglomeration of loosely coupled tools (e.g. links to a website) is maintained that has the advantage of being tailored to the teachers needs, but has the disadvantage of not being consistent (e.g. in look and administration). As a result the existence of synergy effects between different departments is obvious, but the exchange of tools is difficult [111].

CEWebS, developed as a solution, is basically a Web-Service (SOAP) aggregator, that allows to subscribe to learning modules that are distributed throughout an organization. It motivates the faculties

- to develop specialized tools to support teaching and learning (Mathematics, Physics, Computer Science,).

- to provide their tools in a standardized way, so that they are easy to reuse in existing learning platforms/websites.

Objectives of CEWebS are listed [111]:

- Keep it simple: The creation/adoption of new/existing tools is very easy.

- Technological Freedom: Certain (common) interface (WSDL) and protocol (HTTP); write components with an arbitrary programming language.

[2] http://www.blackboard.com/
[3] http://www.moodle.org

- Scalability and Flexibility: Shareable “Software as a Service” components.
- Open Source: Everyone is free to use/create services.
- Patterns: Reuse of best practice scenarios (e.g. course settings).
- Embeddable: Components could be embedded in already existing systems (e.g. other LMS).
- Interaction: Components share data among each other in a transparent way.

5.1.3 Solprov Query Interface

The goal of this query interface, SolProv (Solution Provider), is to allow the user to specify her query in form of a natural language description of the problem statement [112]. SolProv delivers a list of ranked N2Grid-URLs, which provide solutions for these problems, by mapping of problem ontologies (built from a problem space with typical heuristic solutions approaches), to solution ontologies (built from known NN solutions). SolProv is designed and implemented as a standalone OWL-based registry for Web Services (WS) providing mathematical solutions by artificial NNs. Although it is meant as a proof-of-concept it is fully implemented. The SolProv interface can be reached at[4].

5.2 Application Domain: Computational Intelligence

CI is a relatively new research area which focuses on the development of approaches for problem solving mimicking nature [113]. Basically the CI consists of three specific areas of biologically motivated IT: artificial NNs, fuzzy logics, and evolutionary algorithms. CI originates from Artificial Intelligence (AI) by the frustration that AI approaches proved limited for many problems and follows a quest for using nature inspired approaches along the lines of “anything goes”. Significant areas of CI are machine learning (including in particular symbolic multi-strategy and cognitive learning), Web intelligence and semantic web, agents and multi-agent systems, and modern knowledge-based systems [114].

Key application areas of CI are entertainment and gaming, software engineering, business, finance, commerce and economics, knowledge-based and personalized user interfaces [114]. The advent of modern technology in daily lives has urged researchers to seek for a collaborative and scalable resource sharing platform. The CI community has flourished over the time but with few world-wide collaborative environments only (e.g. IEEE CIS). This situation produced an urgent need to form a resource orchestration on worldwide basis. In this collaboration resources are not only hardware and software but also humans who are expert in their respective domains. VOs are generally identified as solution to this problem. Although, this approach roots back to the start of distributed computing, still there are no standard methods to build and maintain VOs. The wide spectrum of CI disciplines can be a

[4] http://big.pri.univie.ac.at:8888/solprovFE/

hurdle to produce such a collective resource pool for CI community. This motivated us to propose a generic collaboration platform to CI community.

5.2.1 Existing Efforts

Until now there are only few efforts visible in this regard. One noteworthy project is the CIML portal (accessible at[5] for machine learning and CI. CIML stands for Computational Intelligence and Machine Learning, a virtual community for providing resources to researchers, students, and general public in the area of CI. It is maintained by the University of Louisville and George Mason University. It supports the effort of building the CI and Machine Learning Virtual Infrastructure Network and is sponsored by the National Science Foundation (NSF) [115].

As the fields of CI and machine learning mature, there is a growing need to provide researchers with the ability to exchange information, share resources, discuss problems and new directions, and learn about other's work. The limitations of traditional scientific communication inspired to create a CIML virtual community, a portal to gather research, education, and application-oriented resources residing that are linked from the CIML site. The goal of the community is to create a place where scientists, students, and the general public can work together despite any of their geographic limitations. Anyone who is interested can share research, obtain resources, or simply learn more on CI.

The CIML portal is, besides its goal to gather all type of interested users, just a static pool for knowledge resources. It totally lacks other forms of resources and new computing paradigms for support of collaborative work. Further it is built without clear design principles. It follows a conventional approach for building the environment without giving clear messages for the IT infrastructure. This situation leads to the problem getting the necessary motivation by the community to contribute. Thus the CIML portal lacks acceptance.

To omit this problem we proposed a blueprint for the design of a Virtual Organization for Computation Intelligence, which we call VOCI [3]. Also as described in E-learning section, the inclusion of CI as a course in curricula, give rise to the need of collaborating resources in this domain [2] [4].

5.3 Virtual Organization for Computational Intelligence (VOCI)

This section provides mapping of N2Grid onto a standard VO blueprint as described in the section 3.4.

5.3.1 Requirement Analysis of N2Grid

Firstly Phase-I is being applied to N2Grid and answers are detailed to the presented questions.

Q1: Why to form a VO?

[5] http://www.cimlcommunity.org/

This question can be answered keeping two aspects in mind, technical needs and social aspects. For technical aspects the purpose is to:

- Share NN objects, data and information worldwide. Provide for a better and efficient solution to the NN problems in an easy but authentic manner.
- Driving stimulus for development is the exchange of information and resources between researchers. This principle is just as valid for the neural information processing community as for any other research community [95].
- Enabling more effective and seamless collaboration of dispersed communities, both scientific and commercial.
- Enable large-scale applications comprising of 10,000 computers, large-scale pipelines etc.
- Transparent access to "high-end" resources from the desktop.
- Provide a uniform "look & feel" to a wide range of resources.
- Location independence of computational resources as well as data.

For social aspects forming a VO is to bring the people together who are common in some respect. Sometimes goals unite the people and sometimes problems bring them closer. So building a community with group of people having problems and those who have solutions can be achieved in form of a VO. A trusted platform to share their commonalities in terms of knowledge, information, applications and procedures makes the face of a VO well recognized and accepted.

Q2: What is the motivation behind participation?

Why should other persons, institutes and/or service providers want to participate in VO? This is the key question, which discovers the needs of participating entities in a VO thus defining the problem domain. Identification of common needs has an important impact on the shape of a VO. For example, specific reasons for participation of the connectionist community are:

- Still no standard simulation NN systems exist.
- Creation of VOCI will inspire other research institutes to collaborate and participate in this VO with their specialized resources.
- Existing systems lack a generalized framework for handling data sets and NNs homogenously. During the training phase and the evaluation phase of a NN the user has to feed the network with large amounts of data. Conventionally data sets are mostly supported via sequential files only and the definition of the input stream, output or target stream of a NN is often static and extremely complex [94].

Socially, being part of an organization related to task performed gives a sense of satisfaction to individuals. The CI research communities are growing on daily

Table 5.1: *Components of N2Grid qualifying for VOCI*

S.no	Components of VO	N2Grid	Missing
1	Common Interest	To serve CI community	N
2	Users	Thinclient, Java Applet	N
3	Tools	Simulation services	Workflow, Provenance tools
4	Data	Data Archives (e.g. OGSA-DAI)	N

bases. Each human who has access to the Internet is a member of a group. It can be a mailing list, email, chat room, social network, professional organizations, E-learning groups and many more. Creation of a VO for the CI community can inspire the people and organizations to become a part of very first virtual informational exchange platform and will motivate them to feed it with their contribution in from of resources and expertise.

Q3: What services are offered by a VO?

Currently, services provided by the N2Grid system are simulation services, uploading data as input file, saving results, teaching and research material, tutorials, presentations, example problems, paradigm selection and addition of new paradigm by authenticated users. A key element is also the querying of proven solution approaches of the community to open problems of the users.

Q4: How are these services fared?

Currently, N2Grid resources are free of cost. A Business Model is foreseen to be introduced to set the usage cost [116]. Also users can provide new paradigms and get benefited from selling the software as service.

Q5: Who are the intended users?

The intended communities are CI research scholars or institutions, and commercial or official organizations utilizing CI resources for their specific tasks. Intended users can be students, scholars, professionals and any person who requires the resources according to NN related job.

Q6: What is life of (membership of) a VO?

N2Grid is an operational system which is intended to last long. It is not created for a specific period of time. Also it is open for updates and improvements from authenticated users. Users are free to participate and to leave.

5.3.2 Phase II: Components Identified in N2Grid

Secondly, the identification of building blocks is an important factor. It gives the list of entities which eventually are being collaborated to form the structure of a VO. Table 5.1 summarizes the components of N2Grid qualifying for VOCI.

Table 5.2: *Gap Analysis - Comparing N2Grid to other VOs characteristics*

S.NO	FEATURES	N2Grid[117]	LEAD[54]	NEES GRID[118]	BIRN[53]	GEON[59]	TWGRID[119]
A	PORTAL INFORMATION						
1	Introduction	Y	Y	Y	Y	Y	Y
2	Team members	Y	Y	Y	Y	Y	Y
3	tutorials	Y	Y	Y	Y	Y	Y
4	News/updates	Y	Y	Y	Y	Y	Y
5	FAQs	Y	Y	Y	Y	Y	Y
6	Contact	Y	Y	Y	Y	Y	Y
B	RESOURCE INFORMATION						
1	Glossary	Y	Y	N	Y	N	Y
2	Publications	Y	Y	Y	Y	Y	Y
3	Presentations	Y	Y	Y	Y	Y	Y
4	Links to related resources	Y	Y	Y	Y	Y	Y
5	Software/Applications	Y	N	Y	Y	Y	Y
6	Technology architecture	Y	N	N	N	Y	N
7	News letter	N	Y	Y	Y	Y	Y
C	RESOURCES						
1	Calender	Y	N	Y	Y	N	Y
2	Blogs/Forum	Y	Y	Y	Y	N	N
3	Research communities/Sites	Y	Y	Y	Y	Y	Y
4	Workshops/Seminars	Y	Y	Y	Y	N	Y
5	Chat/Email	N	N	N	Y	N	N
D	USER INFORMATION						
1	Sign in	N	Y	Y	Y	Y	Y
2	Create account	N	Y	Y	Y	Y	Y
3	Forgot your password	N	Y	Y	Y	Y	Y
E	TASK INFORMATION						
1	Data search	N	Y	N	Y	Y	N
2	Graphical result display	Y	Y	Y	Y	Y	Y
3	Adding services	Y	N	N	Y	N	N
4	Portal Usage statistic	N	N	N	N	Y	N
5	tools	N	Y	Y	Y	Y	N
6	Work flow tools	N	Y	N	Y	N	N
7	Provenance tools	N	Y	N	N	N	N

5.3.3 Gap Analysis

We carried out a detailed comparison of N2Grid with some existing VOs. The comparison chart is presented in Table 5.2. It was very helpful pinpointing the missing components and to decide which parts require improvement and in which regard.

5.3.4 Outcomes and Improvements

In the context of VOCI, we introduce the term *Subject* instead of user or resource. For example, there are some computational resources, data resources, software and hardware resources. But with the introduction of expert's opinion we have a logical being or a human as resource. Also some users are professionals from a specific problem domain. Some users can contribute tools to the VOCI thus acting as resource in this VO environment. So for those resources which are being used in form of a paradigm donor, opinion giver or helping problem solving activity can be considered as Subjects. Today we see other examples of Subjects as members of social networks or informal VOs like Facebook[6] or Myspace[7]. Members not only use the resources of the network but also contribute back by creating different tools/applications and making communities larger by different promotional acts.

After the detailed analysis of the N2Grid system as a candidate for a VO, the following improvements are suggested:

- The N2Grid system currently lakes tools regarding *workflow and provenance*. For any VO these tools are very important to provide a track of problem solving activity. Also workflows have to be saved and reused again, which can reduce the time for problem solving activities. Provenance tools display the history of workflows that created the data, specify each process that was involved, define the origin of the original data and estimate the quality of the data produced.

- Some VOs (e.g LEAD [54]) generates a *graphical view* of the workflow of problem solving activity. This functionality is very helpful for introspecting how a problem gets converted into a solution following which execution path of workflow. The quality of data products is also an urgent question.

- The *expert knowledge* is one of the most important components of a VO. The problem solution activity is made easier and reliable by providing constant help from problem submission till results produced. It increases efficiency level dramatically by helping to validate the problem solving activity in process and to give confidence that components have been accessed in a right manner. It also helps in saving time of the customer who wants to use the VO for a desired action. Also results can be discussed to have a satisfactory level. A mechanism can be developed to get a satisfaction gradation of the computed results depending upon expert's opinion.

- *Ownership users*: The concept behind this novel extension is that sometimes users have only test data and no sample data for training. If a user comes up with a problem for which N2Grid has already been trained, the owner of that model can sell her knowledge (training and evaluation data) to the user to save her training time. This can be viewed in detail by applying any Business Model to provide the proof of concept. Additionally this trained network can provide a list of information to the users about the patterns on which was already trained. Interested clients can have a quote depending upon the data size used for training, time and computational cycles consumed.

- *Business Model*: Currently N2Grid simulation resources are available free of cost. From a VO's context a Business Model must be present in order to regulate the resource usage. User and resources can be categorized according to different criteria and cost can be set according to the role played. A role based

[6] www.facebook.com
[7] www.myspace.com

Business Model can play an important role. It can work dynamically which is changed as the user or resource adapts its role in the VE. This Business Model can also depend on the platform used for communication or as infrastructure. For example Amazon EC2 cloud would charge the users on a set pattern and any VO utilizing this platform can set the cost accordingly. Elements of such a usage calculator model are:

  - Definition of the roles,
  - Costs of roles, and
  - Discounts for the users contributing to VO, while they need some resources.

- *Usage Calculation* is an important tool which helps to keep track how many resources were used. Owner of VO can benefit from this tool while expanding and improving the existing environment. This can give an insight to the number of people utilizing the resources presented by VO. Graphs can be made for a certain period of time to see how the VO is getting attention over a certain period of time.

- *Layered Architecture*: Currently three layers are mapped on grid infrastructure by the N2Grid system. Extending the concept to visualize N2Grid as VO requires few additions in the existing layers. A sub layer concept can be introduced. The knowledge layer should have a tool sub-layer which contains analysis and visualization tools. If N2Grid is extended to include an expert opinion module it will also be added to the knowledge sub-layer. The information layer should also be extended to store workflows. Sharing and reusing workflows is an important activity in VO.

- *Conceptual Improvements*: A VO is simply stated a logical orchestration of resources to achieve a common goal. Logical because resources collaborate on their logical needs for getting a physical or logical solution. Resources can be any entities participating in the problem solution activity. We propose to introduce the new term *Subject* instead of the commonly used term *Resources*. For long time, IT audience is familiar with terms User and Resource in a subject/object manner. User is a subject which uses the resource as an object. With the advancement in user's capabilities and power of resources this definition is not being followed strictly any more.

5.3.5 Concrete Model for VOCI

VOCI is a proposed platform for CI community. The idea is to present an architecture for problem solution activity initiated by a user in the specified domain. This section presents a concrete architecture for N2Grid evolution as a VO.

Figure 5.1 presents the realization of generic frameworks as a NN domain specific VO. Proposed realization is based on N2Grid [94], CEWebs [120], VINNSL [108] and SOLPROV Query Tool [112]. Our ideas of supporting VOs in the CI manifest focus around a modular system developed at Institute of Knowledge and Business

Engineering of the University of Vienna, called CEWebS [120]. CEWebS stands for Cooperative Environment Web Services and realizes a distributed architecture that facilitates short development cycles and the ability to move new functionality to courses as well as research groups very quickly.

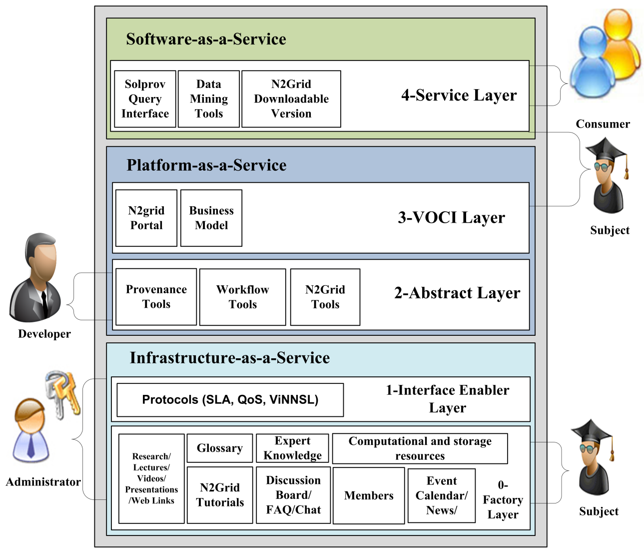

Figure 5.1: *N2Grid based instance of VOCI*

- *0-Factory Layer*: This layer contains domain specific resources, in VOCI context, NN resources. N2Grid Data and paradigm archive services are currently providing the archiving services. High speed distributed nodes provide the computation and storage resources for processing the N2Grid problems. News and event calender keeps user updated regarding portal and research evolutions in current system. Lectures (audio/video/text), publications, presentations, related sites for further information, glossary, tutorials for N2Grid system and access to digital libraries and online journals are part of this layer. Expert knowledge is available both in form of humans and stored information in form of discussion board, chat and FAQ. Subjects have access to this layer. They can contribute here both in hardware and software resources. Administrators are playing their part by managing all the technical activities regarding management and maintenance.
- *1-Interface Enabler Layer*: Provides abstraction in terms of API and protocols to access factory level resources and their utilization in problem solving activities. The combination of N2Grid [94], ViNNSL [108] and CEWebS [120] make this possible in current context. Only administrators have access to this layer.
- *2-Abstract Layer*: In current state this layer contains N2Grid simulation system to create, train and evaluate NN problems. A graphical environment supports problem solving activity. Input, processing and output interfaces are kept simple and consistent. Provenance and workflow tools will be added to the system

in near future. Currently, CeWebS fills the gap by providing a workflow and provenance environment. Developers, from different participating entities in VO contribute to this layer. They enhance the existing modules and introduce new techniques to increase functionality. Bottlenecks in the modules are reviewed and new versions are added to the VO. Here for N2Grid system research students are serving the purpose.

- *3-VOCI Layer*: Consists of N2Grid portal and proposed Business Model. This also gives an entry point to the subject to access the resources. Currently, N2Grid resources are available free of cost. A Business Model is proposed based on roles of users. This model will be added to the model in near future and will also be helpful to build an authentication mechanism for users. CEWebS platform is providing authentication facility at the moment for N2Grid administrative, subject and developers. Figure 5.2 shows N2Grid portal, accessible at [8].

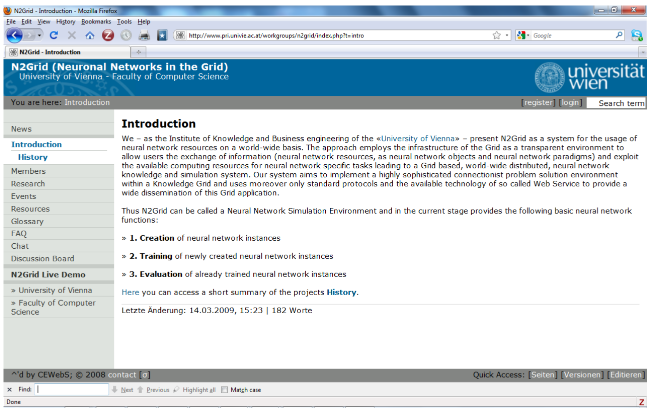

Figure 5.2: *N2Grid Web Portal*

- *4-Service Layer*: provides an alternative way to the user of N2Grid, a “Google”-like interface to query the N2Grid infrastructure on NN resources as solution to given problems. ’SolProv’ is a set of web service based functions to query or update an ontology containing the descriptive properties of several web services providing artificial neuronal networks. [112]. SolProv is detailed in Section 5.1.3. Service Layer also contains Data mining tools.These tools collect results from processing and return the desired output to the user. These tools are helpful in analyzing and deducing knowledge from results obtained. N2Grid system enables user not only to view their results in graphical format but facilitate them to save these results. N2Grid software is also available to the consumers. It can be downloaded to local machine and can be executed to solve a problem. Consumers have access to this layer.

[8] http://www.pri.univie.ac.at/workgroups/n2grid/

5.3.6 User Role

In N2Grid environment user roles are illustrated below,

- *Subject* includes experts and research students in current status. Experts include professors and teachers who are member of N2Grid forum and can be accessed via email. They initiate ideas for new researchers and motivate them to exploit the platform in effective manner. Research students are contributing and utilizing the N2Grid platform for their research work. N2Grid itself, ViNNSL and SolProv are examples of such contributions.

- *Consumer* consists of students at bachelors or master level who can utilize N2Grid resources during their course of studies. To provide access to these resources E-learning systems can become a part of VOCI to benefit students at different level. As a proof of concept CEWebS integrates the N2Grid to provide access to the students to NN resources.

- *Developer* role is played by research students at the moment in the N2Grid environment. With the evolution as VO it is expected to serve the purpose for higher professionals from industry.

- *Administrator* for N2Grid is group of technical experts and students who are working on enhancement of the system. They are responsible for keeping the system consistent and perform all the activities which are necessary to maintain the un-interrupted services to the world-wide NN community.

5.4 Application Domain: E-learning

E-learning, or electronic learning, has been defined in a number of different ways in the literature. In general, E-learning is the expression broadly used to describe "instructional content or learning experience delivered or enabled by electronic technologies" [121]. E-learning is emerging as a collaboration environment to support students specifically in education and humans in general to acquire different skills beyond the geographical limits. E-learning systems are an integrated part of the work environments (education, industry, health and defense etc) [122] [123]. To compete in today's internet-based society, almost every college and university offers an online-based option of study whether it is a complete university experience, an entire degree program, specific course offerings, individual course sections, or web-based components used to enhance face-to-face learning. Effectiveness of E-learning is measured with respect to mode (asynchronous and synchronous), technology acceptance, individual learning style and previous knowledge of computers (computer based E-learning) [123].

Technological advancements change the way how information is shared. With the introduction of next set of buzz words (e.g. ubiquitous learning, mobile learning, blended learning) it becomes necessary to consider how role of user can be improved to keep them moving with the shift. E-learning systems focus more on the interface and utilities provided to the end user. Teachers and students are the intended users of the E-learning system. Very often E-learning systems are seen as burden in the educational process; with the result that such systems do not ease and support but interfere and even endanger the educational process. The restricted design hinders the popularity of the system. Current E-learning systems focus on graduate and undergraduate level students. Research students are essential part of the education but they are not explicitly represented in the existing E-learning system.

E-learning environments exist in different categories and shapes on the face of electronic communication. In distributed computing utilities are implemented as Web Services. In the last few years the CI community is growing fast. This is also reflected by the introduction of CI topics into many IT curricula. Prominent examples are Universities in Germany and Austria, as Bonn, Dortmund, Erlangen Karlsruhe, etc. to name only a few. Also at the University of Vienna, CI topics got importance in the *Computer Science* curricula and in courses as *Knowledge Engineering, Algorithms, Theoretical Foundations*, etc.

This situation produced an urgent need to support the course work of curricula by an E-learning environment which on the one hand specifically focuses on CI content and on the other hand provides a CI guided approach too. As solution we proposed the development of a Virtual Organization for CI supporting E-learning (VELOCI) [2].

This situation motivated our research to design and develop a new form of E-learning system based on guiding principles of VOs. We do not place the user in front of the system, but we inherently integrate the user, student and docent/teacher alike, as constitutive component of the system. In this endeavor we develop the notion of Subject, which comprises all form of roles a human being can carry in such a VO E-learning system, as consumer and provider of (E-learning) resources, and even as resource itself.

5.4.1 Existing Efforts

Efforts have been done previously to define stakeholders in specific to the higher education again targeting the graduate and undergraduate level only [124]. Wagner has also elaborated the motivations and concerns of the stakeholders in detail. Role of research students is not elaborated here also. E-learning is also presented as growing industry [124] [125]. According to some authors E-learning is losing its popularity in favor of new technology pervasive, ubiquitous, mobile and blended learning [126]. Time, Money and efforts have been brought forward to add more utilities. This also brings up the challenge on the user's part to have more skills before using an E-learning system [127]. Teacher is now connected online instead of being in classroom where a limited number of audiences is present. Virtual training demands more technological sophistication both from teacher and student's perspective. Wahlstedt [128] has also elaborated conception of stakeholders by presenting teacher in the role of designer.

5.5 Virtual E-learning Organization for Computational Intelligence (VELOCI)

We presented a novel approach towards stakeholders in VO based E-learning environment. Stakeholders were identified from the RAVO point of view. This pattern was extended to the domain of E-learning, where role of user was viewed as Subject (both consumer and contributor). Subject also act as a resource in the environment to pass on the knowledge as an expert of the field or by developing a new algorithm. Skills are viewed as a shared resource in the environment.

Stakeholders were reviewed in the context of VELOCI. Related research efforts were also presented. Discussion established a new design approach for the VO based collaborative systems in general and for E-learning domain in specific. User is considered the central part of system design rather than an external viewer of the entire system. Shortcomings in the existing approach were justified by proposing the novel view of VO based E-learning system.

Subject a new notion for the user in VO based E-learning system is the shift from the traditional role. We introduced the user roles in VELOCI, where attention was given to the research students in the E-learning system. Also, the notion of subject was introduced in our previous work [1] [2] [3] [5]. VELOCI established a proof of concept that teacher and students are both contributor and consumer of the system. Everything as a service architecture of VELOCI presented interaction of user at different layers and justified their presence in certain roles. VELOCI provided the bases for enhancing the idea of this unique view of E-learning system.

Due to its specific characteristics N2Grid presents itself as a perfect basis for the transition and extension towards a VOCI. CI domain is chosen due to lack of resource distribution platform.

5.5.1 Concrete Model for VELOCI

E-learning environments exist in different categories and shape on the face of electronic communication. In distributed computing utilities are implemented as Web services. In the last few years the CI community is growing fast. This is also reflected by the introduction of CI topics into many IT curricula. Prominent examples are Universities in Germany and Austria, as Bonn, Dortmund, Erlangen Karlsruhe, etc. to name only a few. Also at the University of Vienna CI topics got importance in the Computer Science curricula and in courses as Knowledge Engineering, Algorithms, Theoretical Foundations, etc.

This situation produced an urgent need to support the course work of curricula by an E-learning environment which on the one hand specifically focuses on CI content and on the other hand provides a CI guided approach too. As solution we proposed the development of a VO for CI supporting E-learning (VELOCI) [2].

Figure 5.3 represents an E-learning system that provides a platform for the CI society, both for learners and teachers. The whole architecture is presented as XaaS. All building blocks are participating as services in this collaboration.The N2Grid system is chosen as an example to instantiate the idea presented in the generic architecture.

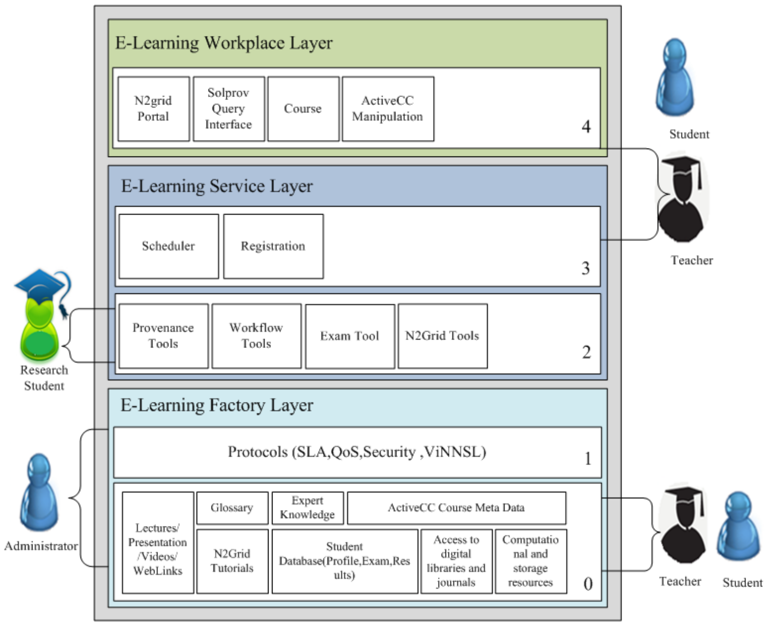

Figure 5.3: *Virtual E-learning Organization for Computational Intelligence*

CEWebS [120] provides the functional support for the E-learning framework.

The specific architecture is composed of three layers, namely the E-learning Factory Layer, the E-learning Service Layer and the E-learning Workplace Layer. Each layer consists of one or more layers depending upon the functionality provided. The E-learning environment consists of three entities User, E-learning Framework and resources used and provided by the collaboration of both in synchronous and asynchronous mode. User roles are explained in the following section, complete details on blueprint and concrete model are available in [2].

5.5.2 User Role

The proposed E-learning architecture supports different types of users namely Student, Teachers and Administrator.

- **Student** Two types of students currently use this system, *Bachelors/Master* and *Research* students. This categorization helps to decide the right of access to the system. Bachelor and Master level students are normally required to take courses, submit assignments, perform exercises in class room, attend and pass exams. These activities are common to Bachelor/Master students. They basically "use" the system, and are therefore present interaction with E-Learning Workplace layer. Research students access this E-learning system for material in specific domain. They are given membership in order to provide access to the online resources. They contribute to the system by developing new tools, upgrading the existing applications and research contributions at E-learning Service layer.

- **Teacher** The teacher is responsible to assemble the E-learning Workplace by putting together a set of services found in the E-Learning Service layer, according to the didactic method used in this course. The teacher is also responsible to give access to the system to a set of students attending a particular course thereby providing membership, because some online resources offered by the system needs login. Two types of teachers defined in this context are Lecturer and Tutor/Teaching assistant. The lecturer is responsible to deliver the materials, which are used by the tutors for guiding students. In some situations, a lecturer guides students as she has planned the course. The tutor is responsible for following actions:

  – Informs students about the course updates.
  – Arranges meetings for problem solution and discussions both online and offline using email, mailing lists, messenger or even social networks (e.g. Facebook) can be used for the purpose.
  – Provides technical support during lecture (multimedia, stationary, audio/visual support).
  – Books lecture halls.

  The lecturer is responsible for course description, setting goals both on teacher's and student's part, preparing lecture material and delivering in classroom. Lecturer and tutor both are in contact to deliver the course contents as efficiently and effectively as possible. Both contribute to the resources at E-learning Factory layer.

- **Administrator** This denotes a single or a group of people who are responsible to maintain the system resources on the E-learning Factory layer. Administrators keep track that virtual orchestration of all the entities are in equilibrium. They do not contribute to the E-Learning Factory layer rather they maintain the necessary infrastructure.

5.6 Application Domain: Computational Science

Computational Science (CS) is striving hard to provide answers for the grand challenges from its sub-domains. It finds its applications in natural science, social and behavioral science, applied science and formal science and other areas which are formed by mixing two domains. CS is defined by the PITAC report [129] as a multidisciplinary field which fuses three distinct interdisciplinary problem solving elements: algorithms and modeling and simulation software, computer and information science, and computing infrastructure.

Thus CS problem solution approaches use concepts and skills from the disciplines of science, computer science, and mathematics. The algorithm development phase combines computational scientists and mathematicians. This phase produces a mathematical model of the problem to be solved. Depending upon the nature of the problem researchers can create new algorithms, modify existing algorithms or get benefit from already developed algorithms. These algorithms are evaluated for their accuracy throughout the modeling process. The next step is to access the data and appropriate computational resources and simulation environments to verify the model being developed. These algorithms require high performance computing and specific architecture. Here, computer science plays its role by providing the infrastructure for executing algorithms and simulation environments to get the desired results. Researchers get benefited by CS techniques in performing experiments that are:

- Dangerous to be executed in lab (e.g. behavior study of a new drug in human body). This allows them to reduce, but not eliminate, the number of animal tests that might have been done prior to the development of these computational pharmacology techniques.
- Too short or require longer time periods. For example models of global climate change allow environmental scientists to run predictive models many years into the future, looking to determine how past, current, and future human endeavors might impact on the temperature of the Earth.
- Costly involving large expenses and equipment of sensitive nature. Especially in chemistry, there are a number of experiments that require expensive instrumentation. Some of these can now be simulated using computational versions of that instrumentation. While this does not replace the importance of having the actual instrument, it does provide the scientist, and the science student, with a way to interact with the instrument. In other areas flight simulators are a good example of the use of simulation software as a cost-saving method. Flight simulators are significantly less expensive than the actual airplane, and are also safer for the pilot.
- Only solvable using computational approaches. Many topics in astrophysics, such as galaxy formation, cannot be observed easily, and certainly are not subject to experimental techniques. Computational models, based on well-understood mathematics, allow the astrophysicist to test a wide variety of parameters and scenarios.

The CS problem solution process involves collaborative efforts from computer science and mathematics. Currently available resources (logical, physical) are not used to their maximal possible extent. Another addressable issue is the absence of an IT standard framework for integrating required resources. At present, there exist no such collaborative platform which provides algorithm development, data access, computational resources and simulation softwares for problem solving activity.

5.7 Virtual Organization for Computational Science (VOCS)

A VO for CS is a solution to this problem. Thus we propose an environment which not only provides computing and simulation resources but also an expert guided research environment for modeling and algorithm development. The specific asset of our framework is the integration of the SPI service model as structuring skeleton of our approach, which categorizes three types of services, SaaS, PaaS and IaaS.

5.7.1 Architecture of VOCS

Figure 5.4 shows the activity flow in VOCS. An Actor represents the stakeholder in VO. It can be of any role (*Subject* or *Enabler*). Query initiation is taken as an event which is processed by the query interface. The query interface supports both remote and desktop based queries. A query is processed by the Virtual Organization Trust (VO Trust). VO Trust is a collection of processes which communicate to process the query. It consists of following sub-components: Authentication and Authorization, Business Model, Contract, Policy, Goal, Feed Back and Resource Information. VO Trust verifies a request from the user by an authentication and authorization system. After the status of member is verified two paths are available. VO Trust has a goal and it can be achieved by either adopting a Business Model in case of a profitable organization. For non-profitable organizations VO Trust has a contract which is based on s policy or strategy to achieve a goal. A feedback component is attached to both Business Model and policy to improve the strategies to get the desired results. The resource information provides information about all the resources available. VO Trust collects this information in the context of the query specification. VO Trust is connected with a provenance tools which utilize the workflow tools to log each activity. The provenance section stores the records of previously solved problems information. VO Trust consults the provenance management to check whether similar problem has been executed already. In this case, existing information is incorporated to reduce the time and effort to process the query. Workflow tools log all the activities and stores workflows for reuse. Query specification and required resource information is passed over to the Technology Architecture. The Technology Architecture features depend upon the underlying platform used for collaboration (e.g. SOA, grid, cloud computing, and Web 2.0). No matter what technology is selected for collaboration it must support the following proposed components: Resource Management (selection, manipulation and aggregation), Security, QoS, SLA and standard protocols for secure communication. Secure communication is a critical issue because in the domain of CS the data processed can be of sensitive nature. For example, applications for defense, aerospace, medical require extra care in processing.

The Technology Architecture contacts the Factory layer to aggregate the resources to solve the problem. The Factory Layer has a resource catalogue which provides information about all the registered resources available. Resources available in CS domain are elaborated here.

- Domain Expert: An expert from Physics, Chemistry, Biology, Mathematics or any targeted area
- Databases: CS problems require access to large databases which otherwise a problem to locate. VOCS provide this opportunity to register databases for different domains and provide required samples according to the negotiation policy decided by participating entities.
- Computational Services: Provides high computing hardware resources for the

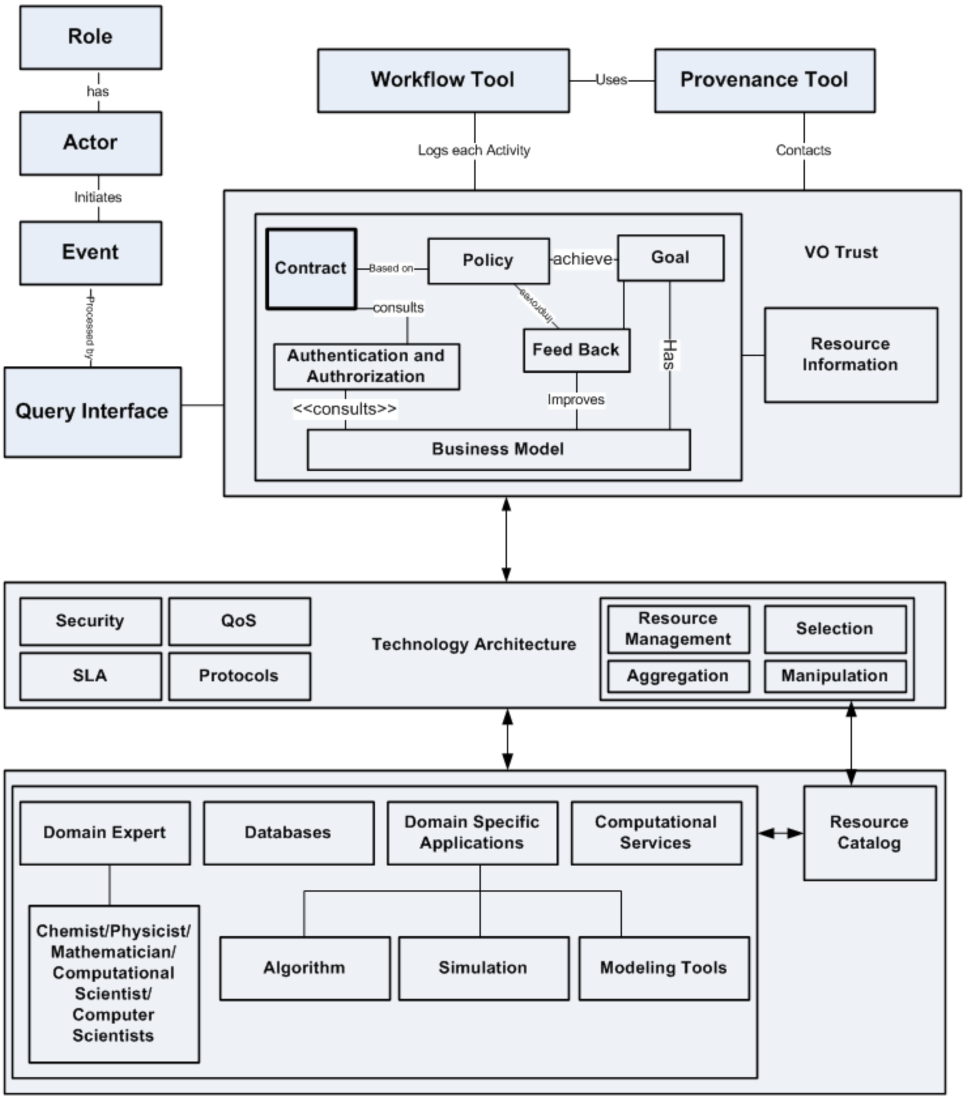

Figure 5.4: *Technical Architecture of VOCS*

computation.

- Domain Specific Applications: This resource is further extended according to the domain selected. In the context of CS, domain specific applications consist of following resources: Algorithm (Problem Solution and High Performance), Simulation and Modeling Tools.

Depending upon the query required resources are aggregated and after processing, results are passed back to the VO Trust through the Technology Architecture. VO Trust sends back the output to the Actor and stores the workflows for the future use. The feedback from Actor is also stored to improve the policies for future use.

5.8 Summary of Research Contribution

This chapter introduced CI, E-learning and CS as target domains for the application RAVO. N2Grid [94] [108], an existing effort is now transformed to N2Cloud [96], was selected as a candidate in the domain of CI. Creation of VO as an evolution of existing system was detailed on the basis of RAVO. Concrete Models are developed for CI, E-learning and CS community.

Relationship between stakeholder and resources is a critical part of VO. In order to meet the shortcomings of the existing systems and demands due to technological enhancements reviewing the stakeholders is a must. Our idea is to present the technology with ease, by integrating user as a resource in the environment. It assigns responsibility of being analytical towards the system usage. We aim to provide a generic pattern for viewing the system as composed of users rather for users. This design is innovative because it gives positions users in an analytical role also. Feedback from the user is an important factor to improve the drawback of the system. Being part of the system in our design, user knows what to improve and how to improve it.

Appendix A provides a detailed development of N2SKY (cloud based system) on the basis of RAVO. A comparison is presented to elaborate how RAVO supports creation of a VO. Next Chapter details the qualitative evaluation of RAVO by a senior researcher. It also provides an insight to N2SKY as a case study.

6 Evaluation

Due to the shortage of time RAVO was instantiated by N2SKY (cloud-based virtual organization for NNs) being developed at University of Vienna by a master student. Details are provided in the following section.

Reference Architecture for Virtual Organization (RAVO) is presented as a standard for building Virtual Organization (VO). It gives a starting point for the developers, organizations and individuals to collaborate electronically for achieving common goals in one or more domains. The RAVO process consists of two parts. First, the requirement analysis phase, where boundaries of the VO are defined and components are identified. A gap analysis is also performed in case of evolution/upgrade of an existing system to a VO. Second part presents the blueprint for a layered architecture RAVO.

6.1 Qualitative Evaluation

Mr. Jürgen Mangler is a senior researcher at Research Group Workflow Systems and Technology, University of Vienna. His expertise include SOAs (including work on mobile devices), with a specific focus on process aware information systems. He developed a light-weight modular process engine to fully support external monitoring and intervention. He further published in the field of RESTful service description, composition and evolution. He was interviewed regarding RAVO as a staring point for developers in NN domain. His opinion about RAVO is detailed below.

Q1. Do you see the need for NN Virtual Organization, allow exploring and integrating NN algorithms as services into your existing workflow?

Answer:

Definitely, NN's are a very good use-case for hosting services in the cloud, as they are quite well understood, and are highly tweak-able through structured limited set parameters. Additionally often they require for extensive training phases on the hardware side are very storage and CPU intensive. Customers may want to save copies of well trained networks and use them over and over again. Thus it covers all the scalability aspects delivered by typical cloud-based solutions. Together with business models for different use cases, and abilities to use, but also provide custom NN services to others (app-store and community for NN services) a thriving ecosystem would be possible.

Q2. Can you give an example scenario, and possible service that would be useful for your area of expertise?

Answer:

Currently we are working in the domain of automatic service selection, based on a set of well defined parameters, such as cost, availability, guaranteed response time, as well as set of fuzzy customer parameters such as location, time and further domain specific preferences. For the automatic service selection algorithm in order to best meet the customer's preferences, it would be highly desirable to employ NN that are trained as the customer continually interacts with the system:

- Assign a NN that is initialized to propose (select) services that the standard customer would select manually.
- Train the NN on the go to adjust to the customer's specific preferences.
- Let the customer specify several selection strategies (risky, conservative) and train NN accordingly.

For a growing number of customers (which each of them have several NN to make suggestions), this demands for a lot of computing power and storage. Currently we are working with a straight forward rule-based system, but tipping into NN services would be highly beneficial for our system. Thus we would require PaaS and IaaS components to host our customer's preferences networks. Combined with a SaaS interface that would allow for efficient monitoring, management and planning of our resources, we could well deliver a much better experience for our customers.

Q3. What business models would be most useful for your use case?

Answer:

A pay-per use system. We would like to scale up our system with customers demands. This means:

- Number of customers that use a NNs.
- Number of NNs that a customers use.
- Usage intensity.

This would allow us to transparently map and pass the costs to the customers.

Q4. Can you come up with additional, related areas, where such a system would be beneficial?

Answer:

Could be beneficial for all organizations that currently use rule-based systems, operate with high numbers of input parameters, are not dependable on the accountability of their system, but instead require good and fast results.

For many areas such as stock market analysis, neuronal networks are currently already in use, but obviously for out-sourcing such applications, special security and encryption requirements are in place. Thus a special focus for different (maybe pluggable) security protocol and encryption mechanisms would be necessary in order for such a platform to become successful.

Q5. Which long term developments for such a system can you imagine?

Answer:

The system could be central hub (i.e. marketplace) for the NN community, including NN services and hosting, but also documentation and consulting. Like for the App-Store the marketplace operator could define a revenue sharing model for content and service providers. With specialized hosting offers (i.e. specialized hardware) geared towards the requirements of NN, the marketplace operator could bind customers and developers. The marketplace operator should at some point, avoid participating in consulting and development, but instead act as a trusted third party between developers and marketplace customers.

Q6. How will you justify Layered approach of RAVO for N2SKY design? Why SaaS, PaaS and Iaas layers are necessary?

Answer:

- SaaS for Querying and Account management.
- PaaS for providing the services.
- IaaS for deploying and running services and hosting the generated data.

Q7. How is a Stakeholder in RAVO seen in N2SKY?(what are the stakeholders in N2SKY)

Answer:

Stakeholders I could foresee are:

- Service Providers and Sevice Mashup Providers.
- Service Users
- RAVO operators.

6.1.1 Conclusion

Qualitative analysis of RAVO in the context of NN domain is presented. We can conclude following

- RAVO provides a suitable basis to create a VO for any domain (here specifically NN community).
- Business needs of the community should be addressed by providing a strong basis for integrating Business Models in to RAVO.
- Layered architecture (based on SaaS, PaaS, IaaS) allows the separation of components and services according to the activity performed.

6.2 Case Study

Mr. Erwin Mann, a master student at University of Vienna, chose RAVO as the template to develop a cloud-based virtual organization for NNs called N2SKY. Mr. Erwin Mann has experience in implementing service-oriented architectures (SOAs), service orchestration, to create workflows and in porting such systems to cloud-based environment. N2SKY brings together both NN paradigm developers and users who deal with problems that are beyond conventional computing possibilities.

N2SKY provides a standardized description language for describing neural objects (instances of neural paradigms) called VINNSL. Furthermore N2SKY provides a business model for researchers and students but also for any interested customer. N2SKY's core process is the Simulation Service including creation, training and evaluating of neural objects in a distributed manner in the cloud.

We carried out a detailed interview with Mr. Erwin Mann regarding the process of instantiating N2SKY on the bases of RAVO. We divide the questions in three categories, namely Requirement Analysis, Gap Analysis and Implementation, to depict a clear vision. First category is requirement analysis which details questions to justify the need of N2SKY in the intended community. Gap analysis presents the comparison of the existing system and RAVO to detail the theoretical grounds. Implementation presents questions about how N2SKY was created using RAVO as a base.

6.2.1 Requirement Analysis

Q1. What is N2SKY? What give rise to the need to create this VO??

Answer:

N2SKY brings the former N2Grid into the cloud and includes a business model with different pricing models. The already well-functioning grid infrastructure of N2Grid is placed on the current state of technology by using RESTful Web Services, JSON as data format, HTTP for data transfer and enhanced replication and persistence mechanisms.

Q2. What is the motivation behind participation? Why should other persons, institutes, service providers want to participate in N2SKY?

Answer: Motivation behind participation in N2SKY is to,

- Share neural net paradigms, neural net objects and other data and information between researchers, developers and end users worldwide.
- Provide for an efficient and standardized solutions to NN problems.
- Transparent access to "high-end" neural resources stored within the cloud from desktop or smart phone.
- Provide a uniform "look and feel" to NN resources.
- Location independence of computational, storage and network resources.

Use Case: Breast cancer cell classification A group of cancer researcher and programmers develop a system to search for cancer cells in tissue images made by a microscope. By classifying these cells, breast cancer diagnoses are created using artificial NNs. The user interface component is integrated into the rich client application at the end user's desktop whereas the diagnosis service component is hosted in the cloud on the NN Layer and utilizes an appropriate paradigm offered by the N2SKY Simulation Service. The end users pay for system hosting, paradigm support and computation cycles a flat rate fee on a monthly basis. One part of these revenues will be forwarded the paradigm provider, the other part remains with the University of Vienna as N2SKY operator.

Q3. What services are offered by a N2SKY? What scenarios are supported by N2SKY?

Answer:

- Read and discuss tutorials, research papers and presentations (All stakeholders).
- Publish tutorials, research papers and presentations (Contributors).
- Manage stakeholder account: edit data, select payment method, credit account, cash out (All stakeholders).
- Integrate NN Paradigm into N2SKY and select pricing models to offer (Contributors).
- Integrate any JVM-compliant software component into N2SKY by providing various DBMS (Administrators).
- Integrate hardware into N2SKY, e.g. sensors or scanner and select pricing models to offer (Contributors).
- Query Interface: Search for NN problems and their solutions (Consumers).
- Select resources that will be used and choose a pricing model for them (Consumers).
- Simulation service: Create, train and evaluate neural objects (Consumers).
- Create end user bill: calculate and send bill, debit amount and credit parts of it the contributors (Controller as subclass of Administrators).
- Check stakeholder accounts and send reminders if bills were not paid (Controller).

Q4. How are these services fared? What is the type of the resources/business model? Is there a specific Business model as foreseen in RAVO?

Answer:

A secured trusted platform is the basis of the N2SKY business model. In case of educational purpose N2SKY offers its services for free. User authentication is also required to avoid abuse. N2SKY offers the following pricing models where the price is the sum of cloud provider fees and paradigm provider fees in three different SLAs (premium, standard and minimal):

- Pay-per-use: It is the standard model if no other is selected.
- Flat rate: A fixed monthly fee regardless of the intensity of use
- Local execution: Equal to the flat rate model but without cloud provider fees because the consumer operates the system on his own infrastructure.

- Negotiated roles: Large customers who have special requirements have the possibility to agree to special packages which can then be reused for similar users.
- Dynamic negotiation: If the user wants to use system extensively it is possible to dynamically negotiate the terms of use with both the cloud provider and the paradigm vendor where the negotiations should be moderated by the N2SKY system.

Q5. Who are the intended users? Who will eventually use and get benefited from this N2SKY?

Answer:

In N2SKY, users are defined as:

- Subjects: NN researchers, professors and Master's students - both paradigm providers and end users.
- End users: Researchers, lecturers, students and commercial users that are interested in NN problem solutions.
- Developer: NN paradigm.
- Administrators: System administrators and business administrators. The N2SKY controller is a business administrator that controls the business workflow (administration of resources and pricing models, invoicing, payments, bookkeeping, reminders).

Q6. What is the life of (membership of) N2SKY? Is temporal alliance or permanent participation expected?

Answer:

N2SKY is based on N2Grid, an operational system launched a few years ago which is not created for a specific period of time. Also it is open for updates and improvements from authenticated stakeholders. Users are free to participate and to leave.

6.2.2 Gap Analysis

Q7. What are the existing VOs in the field of Computational Intelligence?

Answer:

To the best of my knowledge there exist only one noticeable VO in the field of Computational Intelligence and Machine learning known as CIML (Computational Intelligence and Machine Learning) portal. This VO portal is an international multi-university initiative. Its primary purpose is to help facilitate a virtual scientific community infrastructure for all those involved with, or interested in, computational intelligence and machine learning. This includes CIML research-, education, and application-oriented resources residing at the portal and others that are linked from the CIML site.

Q8. Why you based your development on RAVO?

Answer:

RAVO identifies components within an extended cloud SPI stack that helped me to choose and append components for a special NN VO. I think RAVO is a better approach than CIML to build VOs especially in the context of cloud computing. A collaboration of N2SKY with the CIML community is also entirely conceivable.

Q9. Is N2SKY an up-grade of an existing system?

Answer:

Yes, the main part of N2SKY is based on N2Grid, especially the NN Simulation services and the Service Registry.

Q10. Was resource categorization in RAVO applicable to N2SKY Resource pool? What logical and physical resources are available in N2SKY?

Answer:

- N2SKY uses computation and storage according to RAVO's physical resources and appends network traffic as a further physical resource.
- N2SKY's logical resources are derived from RAVO and are refined especially for NN purposes.

6.2.3 Implementation

Q11. How the Gap analysis eased the Implementation of N2SKY?

Answer:

It has already done some preliminary work on and I could use these ideas for my work to develop N2SKY. The most important outcomes of the gap analysis are:

- Workflow tools will be integrated to execute micro flows during training and evaluation phases of neural objects.
- Provenance is missing to collect metadata about each simulation run.
- User authorization to access particular resources will be integrated.

Q12. How RAVO eased the development of N2SKY?

Answer:

Theoretical aspects: First part of RAVO, requirement analysis and gap analysis, provide strong theoretical basis to define the boundaries of a VO. The requirement analysis phase provided a clear vision "*Why there is a strong need to have N2SKY*".

N2SKY is an evolution of an existing grid based system, formerly called N2Grid. Component identification and gap analysis was must to find what additional components are required to evolve.

Technical aspects: RAVO presents an SPI-based system architecture model, which meets our requirements for N2SKY. It eased the separation of components in different layers and combining these components in a problem solution activity. These five layers are:

- 0 - Factory Layer: includes physical and logical resources.
- 1 - Infrastructure Enabler: Management of the resources in layer 0: Computation management, Resource management and Network management.

- 2 - Abstract Layer: Components to manage VOs.
- 3 - NN Layer: Components to handle NNs.
- 4 - Service Layer: User interfaces and services especially for end users and paradigm provider.

Q13. How will you justify Layered approach of RAVO for N2SKY design? Why SaaS, PaaS and Iaas layers are necessary?

Answer:

I divided the components for N2SKY into three categories: mandatory, integration phase 1 (IP1) and 2 (IP2) In the Factory Layer I added network components to physical resources (IP1). In the Infrastructure Enabler Layer I added two mandatory resource management components: the Component Replication (to replicate paradigm components for concurrent computation) Service and the Cloud Data Archive (to manage distributed data). Abstract Layer: I moved the business model (IP1) from the VOCI layer to the Abstract Layer and included SLA, monitoring and accounting into the business model. NN Layer (according to VOCI layer): I added two components: the mandatory NN Simulation Service and the Business Management Service (IP1). Service Layer: In addition to the Query Interface (IP1), the web portal (mandatory) and a smart phone app (IP1) I added Hosted User Interfaces (IP2) for Hosted Components (IP2) in the NN Layer as described in the use case in Answer: 2.

Q14. How was the implementation process eased?

Answer:

The helpful during the requirements analysis process of N2SKY. For the concrete implementation decisions I analyzed the former N2Grid system to reuse Java code that meets the requirements and to develop new modules which are not there either, or no longer reflect the current state of the art.

Q15. How is a Stakeholder (defined in RAVO) seen in N2SKY?

Answer:

The stakeholder classification was used to create a role-based user management for N2SKY. User roles do not need to be disjointed and one user can have multiple roles.

Q16. What is Subject (proposed in RAVO) in N2SKY?

Answer:

I think that the concept of the Subject is a theoretical approach which has not been implemented into N2SKY until now. Subject which can be both a stakeholder and a resource is a good theoretical concept but could be reduced as the expertise of a stakeholder (an expert). For implementation purposes we propose a knowledge management system to manage formal knowledge, informal knowledge can be demanded from an expert. For our user management component we use roles where each user can own multiple roles, e.g. end user and paradigm provider.

Q17. What is the business perspective in N2SKY?

Answer:

In N2SKY,

- Contributors have the opportunity to earn money with their services offered.
- Consumers are able to use free or fee-based services offered exclusively in N2SKY.

Q18. How new concepts of Subject was helpful in implementation?
Answer:
I think that the concept of a Subject which can be both a contributor and a consumer and a stakeholder and a resource is an important new concept. Regarding stakeholder categorization our user management component follows this concept by allowing that each stakeholder may have several roles at once. Regarding resource categorization the Subject as subclass of Expertise may be demanded like a resource with a pro-defined pricing model. For example, an expert in developing secure RESTful Web Services may be contacted via e-mail or telephone to Answer: special questions and write an implementation concept. The expert will calculate the customer a fee based on consultation time and hour rate which was announced previously.
Q19. Were the roles easy to implement?
Answer:
Yes, I adjusted the RAVO roles for N2SKY and derived a new role: the Business Controller.. The Business Controller is a sub-class of Administrator and is responsible for the N2SKY business workflow described in answer to Q5. By default the Business controller has unrestricted access to all stakeholders and resources but he can be restricted from the administrator to a particular sub-tree of resources or stakeholders.
Q20. Your view of the architecture as a developer (RAVO and N2SKY)? Your view of the architecture as a user?
Answer:
As a paradigm developer or end user I am only interested in the SaaS layer consisting on a special NN web portal and Smartphone app or other web interfaces or rich clients. If the paradigm meets the N2SKY specifications, the web portal provides functionality for uploading paradigms and the selection of terms of use and pricing models.
Q21. Can you give us the time line for the phases to develop?
Answer:
Theoretical work: June to February 2012.
Implementation: October 2011 to February 2012.
Q22. What technology you chose for implementation? What made you choose this technology?
Answer:
N2SKY Services:

- Java because N2Grid is also Java-based

- Spring framework as a lightweight alternative to EJBs to realize service-based software components. Used features of Spring are Spring MVC and JPA (O/R mapping)
- Tomcat as Web and application server
- Maven2 as project management and build tool. A key factor for Maven is the dependency management in the pom.xml where dependencies (libraries) and plug-in (adapters to external software) can be managed and were automatically downloaded from repositories.
- Jersey as framework for realizing RESTful Web Services. Jersey is the reference implementation from Sun/Oracle for RESTful services.

N2SKY Web Portal:

- Additionally HTML 5 and CSS for the user interface
- Grails framework to realize Servlets and Server Pages. Grails is the alternative to Ruby on Rails to use existing Java code directly (from the former N2Grid GUI) without wrapping it into Web Services. Grails uses Spring and Hibernate in the background, The freely available SpringSource Tool Suite (STS) is the IDE of choice to work with Grails and Spring.

Cloud Infrastructure:

- Eucalyptus: the software platform for the realization of private clouds. It is compatible with Amazon's EC2 and S3 services.

Q23. What alternative technologies would you as an expert also consider?
Answer:
RAVO is not dependant on a specific technology. Developers have an open choice for building a VO. I chose the above mentioned technologies because in my opinion these are the state-of-the-art technologies for the Java platform with an optimal support for Eucalyptus. Alternatively, I can achieve these goals with Ruby on Rails for the web portal or an Objective-C application for the iPhone applications. For the distributed data storage in the cloud also mechanisms based on the Map Reduce design pattern are conceivable.

6.3 Conclusion

Both researcher gave their opinion after critical analysis of RAVO. Mr. Mangler's abstraction of RAVO in terms of "Q&A" is helpful for the developers of VO in any domain. Mr. Erwin Mann has applied RAVO and developing an instance in the domain of NN. We deduce the following statements from this evaluation.

- RAVO best fits needs of community for developing a VO from scratch.
- RAVO supports evolution of existing systems in to a VO. N2SKY is an example of such evolution.

- RAVO is presented in a layered fashion, with a choice of mandatory and optional components. Layered approach make it easy to distribute the components in different layers and also developers are not bound to choose the exact distribution. RAVO is a flexible and extendable framework. Developer can change the components and move them to any desired layer. For example in N2SKY, components have been moved to different layer as compared to RAVO.
- RAVO is not technology dependent. Both the researchers described their alternative choices which establishes the technological independence of RAVO.
- Categorization of resources into logical and physical is a new dimension for VO developers. Inclusion of human expertise as a resource supports the demanding nature of problem solving ability, thereby increasing the level of trust in users.
- RAVO presented a new concept of stakeholder, *Subject*. A unique idea of how a stakeholder can become a resource in a VO. Being consumer and producer at the same time is difficult to implement. RAVO make it easier by introducing the stakeholder categorization.
- RAVO foresees a Business Model which is introduced in N2SKY as a mandatory component. Stakeholder's roles are integrated in Business Model to set the usage and cost policy.

6.4 Summary of Research Contribution

We are thankful for Mr. Jürgen Mangler and Mr. Erwin Mann for their cooperation and time. This questionnaire shows that N2SKY is the proof of concept for RAVO. It saves time and effort for building the VO. N2SKY also incorporates the proposed business perspective which can be a point of interest for business community. Next Chapter presents the conclusion of our research efforts in a graphical form.

7 Conclusion

The diagram presents the activity which is detailed in this thesis in a pictorial form. The activity is explained in a bottom-up fashion and top layer presents the concrete model starting from the concepts and requirement analysis. This can also be viewed as a life cycle of the VO. Each Layer is an input to the layer above. We presented RAVO as a standard to the research community. RAVO was justified by development of concrete models in different domains (CI, E-learning, Social networks, and CS). A detailed comparison of an instance being developed on the basis of our proposed works was also detailed. A complete view of the RAVO is shown in Figure 7.1.

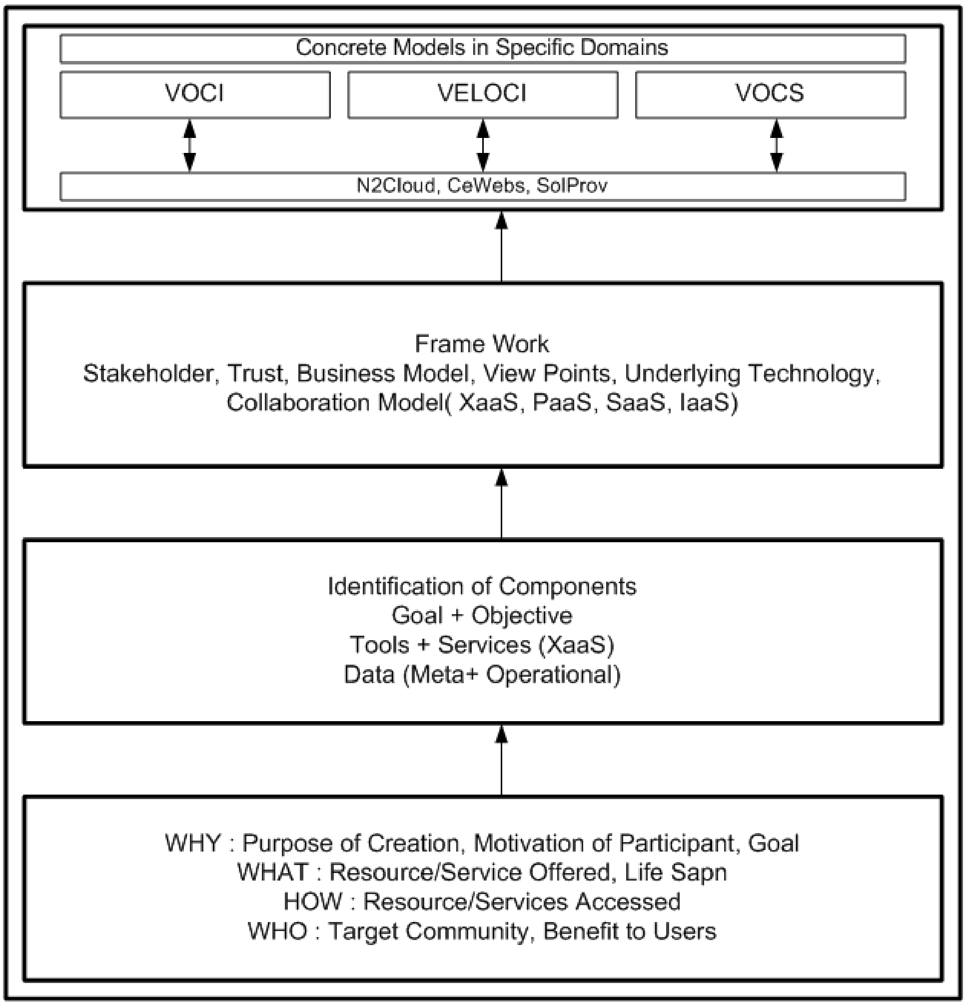

Figure 7.1: *Bottom-up Process of Building Virtual Organization in different Domains*

A N2SKY: An Instance of RAVO

A.1 Introduction

We proposed RAVO as a standard for creation of VO in any domain. RAVO is chosen as a baseline for implementing a cloud based VO for NNs namely, N2SKY. Erwin Mann, a master student at University of Vienna, has based N2SKY on RAVO and has produced a concrete instance out of our proposed standard. This chapter compares N2SKY with RAVO to reveal the process of creation of N2SKY. The comparison justifies and proves how RAVO supported different development phases of N2SKY. We explain N2SKY as an instantiation of RAVO but with concrete components. We divide this comparison in 3 levels. First, *Requirement Analysis Phase* that defined boundaries of N2SKY. Second, *Component Identification Phase* which made it easy to identify the components of N2SKY and also choose between optional and mandatory components. Third, *Implementation Phase* that reveals how technology independence, XaaS and layered distribution of components made it helpful to implement the system. The stakeholders envisioned in RAVO are also implemented as part of N2SKY.

A.2 Requirement Analysis in terms of RAVO

In section 3.4.1 we detailed a series of questions which must be answered by the responsible authorities for creating a VO. N2SKY utilizes this pattern for defining the requirements boundary of the system. These questions are answered in detail in an interview by Mr. Erwin Mann for evaluation of RAVO, which are presented in Chapter 6.

A.3 Component Identification in terms of RAVO

N2SKY is a layered architecture instantiated from RAVO. The N2SKY is shown in Figure A.1. N2SKY is also presented as an XaaS, based on Cloud SPI model. It consists of 3 layers, namely SaaS, PaaS and IaaS. These layers have sublayers similar to RAVO. Each layer has some components which are either mandatory or optional depending upon their participation in VO. RAVO is explained in Chapter 3. Figure 3.3 shows RAVO framework. A detailed comparison of RAVO and N2SKY components is given in Table A.1.

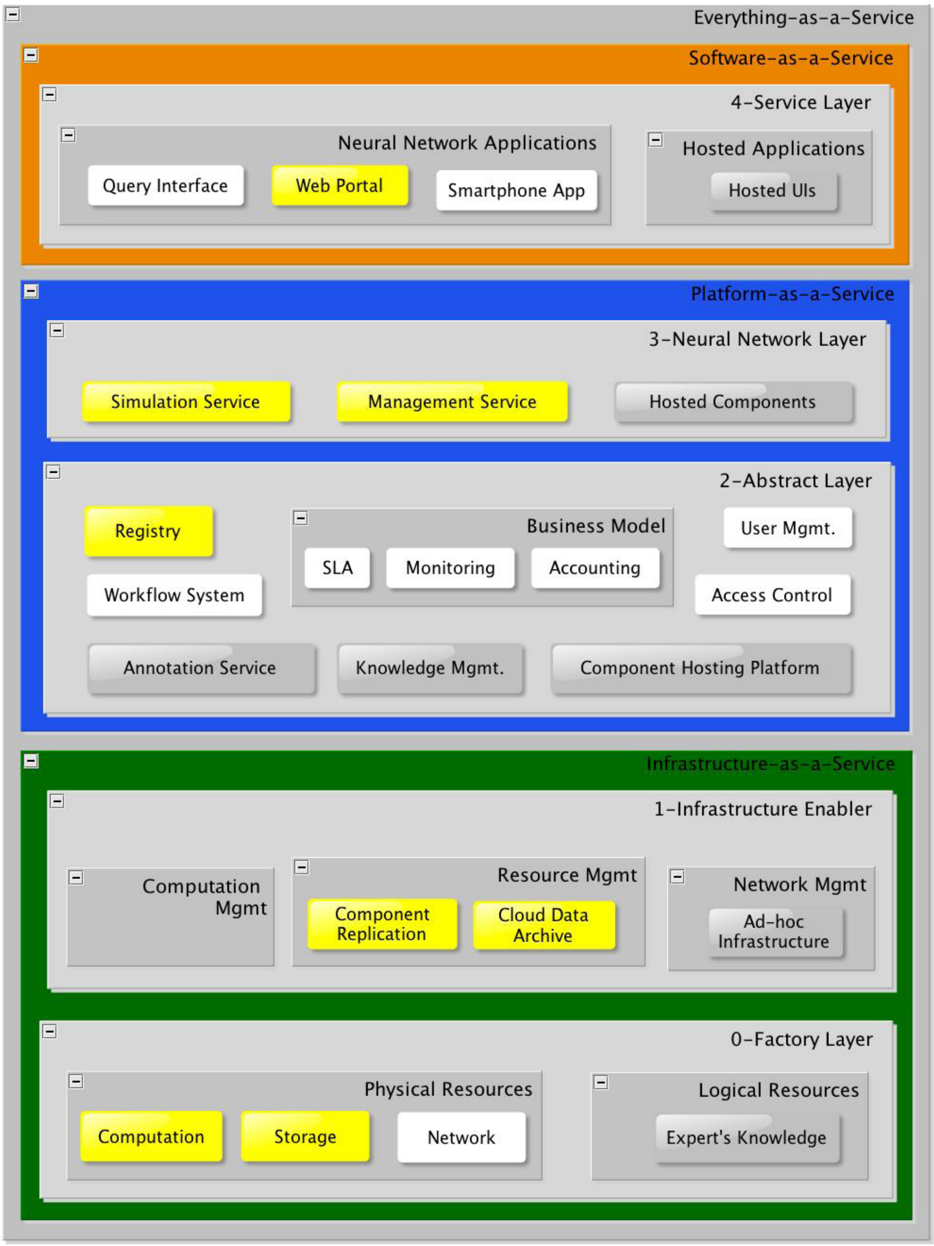

Figure A.1: *N2SKY*

A.4 Interface Specification in terms of RAVO

Section 3.3 presented interface specification for components. Here, we analyze how these interface specifications were used in N2SKY. We compare the underlying framework RAVO with its instantiation as N2SKY, in a top-down fashion. We start with SaaS layer:

Table A.1: *Comparison: RAVO vs N2SKY*

Layer	SubLayer	RAVO	M/O	N2SKY	M/O
SaaS	Service Layer	Query Interface	M	Part of Web Portal	M
		Domain Specific Application	M	NN Simulation Services at NN Layer	M
		Data Mining Tools	O	N/A	N/A
PaaS	VO Layer (Neural Network Layer in N2SKY)	VO Trust	M	Management Service, Usermanagement, Access Control Business Model with SLA, Controlling and Accounting	M
		User Interface	M	Web Portal	M
	Abstract Layer	Resource Management	M	Registry, Business Model with SLA, Accounting, SLA	M
		Provenance Tools	M	N/A	M
		Workflow Tools	M	Workflow System	M
		Graphical Interface	M	Part of Web Portal	M
IaaS	Infrastructure Enabler Layer	QoS	M	Included at Abstract Layer	M
		Security	M	Included at Abstract Layer	M
		SLA	M	Included at Abstract Layer	M
		Technology Architecture	M	Ad-hoc Infrastructure	M
	Factory Layer	Resource Catalogue	M	Management Service and Registry	M
		Expert's Knowledge	M	Knowledge Management at Abstract Layer	M
		Data Services	M	Data Archive at Abstract Layer	M
		Computational Services	O	Computational Replication Service at Abstract Layer	M

A.4.1 SaaS Layer Comparison

SaaS layer of RAVO consists of optional and mandatory components. Choice of components and decision on their status (mandatory and optional) is open for the developers. The inclusion of components is dependent on the requirement definition by the stakeholders.

SaaS Layer has one layer, named Service layer. Its components are defined in detail in Chapter 3. Here, only tables are included for the sake of comparison.

- Query Interface: RAVO proposes Query Interface as a mandatory component at Service Layer. Details are shown in Table A.2. In N2SKY, Query Interface is also included as a mandatory component. Existing instance of the Query Interface component is shown in Table A.3. Implementation is planned.

- Domain Specific Application (DSA): DSA is a mandatory component. Details are shown in Table A.4. N2SKY has a simulation service but at Neural Network layer (sub layer of PaaS). N2SKY includes DSA as NN specific applications. N2SKY is planned to include NN specific applications. The Simulation Service provides the creation, training and simulation of neural objects which in turn are instances of NN paradigms. Currently, Simulation Services are provided at NN Layer of N2SKY. Specifications are shown in the Figure A.2 in a tabular form.

- Data Mining Tools: Data mining tools are an optional component of RAVO. Details are shown in Table A.8. N2SKY has not included this option.

N2SKY also has one layer, named Service Layer (similar to RAVO). Extended components included at Service Layer in N2SKY are:

- Web Portal: N2SKY Web Portal is a mandatory component. Existing instances of the Web Portal component Interface is shown in Table A.5.

- Smaprtphone APP: Existing instance of the Smartphone App component is shown in Table A.6.

- Hosted UI: Existing instance of a Hosted User Interface component Interface is shown in Table A.7). Implementation planned.

A.4.2 PaaS Layer Comparison

PaaS layer is composed of two layers, namely VO Layer and 2-Abstract Layer. Component Specification is detailed below. In N2SKY PaaS consists of 3-Neural Network Layer and 2-Abstract Layer.

A.4.2.1 VO Layer comparison with 3-Neural Network Layer

In RAVO VO layer has the following components:

Simulation Service			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	schemaLocation	URL, directory	*
	registryURL	URL	*
Inbound interface		Web Service	*
getServiceDescription()		synchr.	*
OUT	nnDescription	nnDescription	*
createNeuralObject()		synchr.	*
IN	nnDefinition	nnDefinition	*
OUT	submissionID	String	*
train()		asynchr.	*
IN	nnDefinition	nnDefinition	*
IN	nnData	nnData	*
IN	SLA	String {A|B|C}	
IN	callbackURL	URL, unique	*
instant output	submissionID	String	*
evaluate()		asynchr.	*
IN	nnDefinition	nnDefinition	*
IN	nnData	nnData	*
IN	SLA	String {A|B|C}	
IN	callbackURL	URL, unique	*
instant output	submissionID	String	*
checkStatus()	for train(), evaluate()	synchr.	*
IN	submissionID	String	*
OUT	status	int	*
getResult()		synchr.	*
IN	submissionID	String	*
OUT	simulationResult	simulationResult	*
getNNDefinition()		synchr.	*
IN	submissionID	String	*
OUT	nnDefinition	nnDefinition	*
Callback to operations:	train(), evaluate()	Web Service	*
sendResult()		asynchr.	*
IN	simulationResult	simulationResult	*
Used components:	Registry: registerService(), removeFromREgistry()		

Figure A.2: *Interface specification of the Simulation Service*

- VO Trust: Mandatory component of VO, which is responsible for enabling resources, defining policies to achieve a goal. It has several components and is extendable according to the need and requirement of stakeholders. N2SKY has distributed Trust component in to different modules. These components are shown in Table A.9. In N2SKY, Neural Network Layer has a Management Service component to serve the purpose. Details are shown in Table A.10. Other components are available at Abstract layer namely, Business Model with SLAs and Accounting.

- User Interface: User Interface is a mandatory component for solving problem utilizing VO PaaS utility. It provides an interface to interact with the VO. Details are shown in Table A.12. N2SKY also realizes this component as a part of Web Portal.

Extended Component of N2SKY:

- Hosted Component: Provides and interface for components hosting platform A.11. Integration planned.
- Simulation Service: Already described in Service Layer comparison. It is a mandatory component that is part of Neural Network Layer of N2SKY.

A.4.2.2 Abstract Layer Comparison

RAVO and N2SKY both have this sub layer named 2-Abstract Layer. Components of these layer in RAVO and N2SKY are compared.

- Resource Management: Resource Management is a mandatory component of Abstract Layer. It provides a mechanism to select and aggregate resources for a problem solving activity. Depending upon the underlying technology, VO developers can deploy different resource management tools. RAVO being technology independent lists a brief description in the Table A.13. In N2SKY resource management is achieved via mandatory Registry component shown in tabular form in Figure A.3.
- Workflow Tools: RAVO Workflow Tool Interface is shown in Table A.18. N2SKY also have a Workflow System under development. Interface is shown in Table A.19.
- Provenance Tools: Provenance Tools are proposed in RAVO but they are not included in N2SKY.
- Graphical Interface: A mandatory components which facilitates interaction with VO easier and helps user to get results in an understandable format. It also assists user in formulating queries and browsing in VO environment. Details are shown in Table A.20. In N2SKY Graphical Interface is implemented as a Web portal described earlier.

Extended Components supporting VO Trust (as proposed in RAVO) Functionality:

- Controlling and Accounting: This component along with SLA component serves as a Business Model. In RAVO Business Model is optional. Details are shown in Table A.15. Integration Planned.
- Usermanagement: Details are shown in Table A.16. Integration planned.
- Access Control: Table A.17. Implementation planned.
- SLA: Table Details are shown in Table A.14. Implementation planned.
- Annotation Service: Details are shown in Table A.21.

- Knowledge Management: It refers to Expert's Knowledge of RAVO defined at Factory level. Details are shown in Table A.24. Implementation planned.
- Component Hosting Platform: Details are shown in Table A.22.

A.4.3 IaaS Layer Comparison

IaaS layer is composed of 1-Infrastructure Enabler Layer and 0-Factory Layer. This layer froms the fabric of RAVO. All the resources are available in Factory Layer and are exploited through Infrastructure Enabler Layer.

Infrastructure Enabler Layer in RAVO brings an open choice for the developers for underlying technology. QoS, Service Level Agreement (SLA), Security, Fault tolerance and Disaster management are aspects to be considered in particular. Further extension can be done by developers. Interface for this layer is abstracted in Table A.23.

N2SKY also have an Infrastructure Enabler Layer. It contains following components.

- Data Archive: Implemented as a mandatory component of N2SKY. Interface specifications are detailed in Table A.30.
- Component Replication Service: Existing instance of the Component Replication Service Interface is shown in table A.29.

Factory Level of RAVO is also instantiated in N2SKY. It has following components in RAVO

- Resource Catalogue: Resource Catalogue module is an extension of Resource Management Component. It is a mandatory components. It keeps information about resources which is of interest to VO. Details are shown in Table A.25. In N2SKY this task is achieved by Registry component shown in Figure A.2.
- Computational Services: RAVO offers Computational Services as a mandatory component. Interface specification are shown in A.28. In N2SKY this component is realized by Component Replication Service. It is a mandatory component which act as N2SKY Paradigm Archive Service.
- Data Services: This component of RAVO is realized by N2SKY as a part of Infrastructure Enabler Layer.
- Expert's Knowledge: N2SKY implements this component of RAVO as Knowledge Management as a subcomponent of Abstract Layer. The component interface is already described in Section A.4.2.2

A.5 Stakeholder Comparison

Stakeholder defined by RAVO are detailed in Chapter 4. N2SKY extends the categorization shown in Figure A.4. The Users identified in N2SKY are listed below:

- N2SKY Controller: The N2SKY Controller is able to add (+) and remove (-) any role to any user over a graphical user and role management interface.
- Cost Controller: The Cost Controller is responsible for the expenses of a particular cost bearer unit and has all permissions within this unit.
- Developer: The Developer of neural network resources has all permissions within the unit except of the manipulation of Cost Controller roles.
- End User: The End User is able to consume services up to the defined budget limit per month. Budget limit operations have to be approved from him the Cost Controller responsible for this cost bearer unit.

These roles are useful to develop access right and integration in Business Model. Permission according to roles defined in N2SKY are shown in Table A.31. Stakeholder.pdf

A.6 Summary of Research Contribution

This Chapter provided a detailed comparison of RAVO and an instance build on it, named N2SKY. This system is under development and fully utilizing the generic patterns provided by RAVO. Comparison revealed the following conclusion.

- RAVO provides strong theoretical grounds to clear the vision of VO developers and participants before they start building a community.
- Requirement Analysis and Component Identification phases enable developers to list mandatory and optional components. The purpose is twofold. First, must parts of the VO are confirmed. Second, optional parts leave room for future requirements and upgrades.
- RAVO framework is flexible and generic. Components at different layers are moved or integrated with other parts as it eases the developing process.
- RAVO is technology independent and it gives freedom of choosing any suitable tools and programming languages.
- RAVO emphasis on providing graphical interface to ease the end user so that they can communicate and formulate their queries easily. The interface should not be complicated that only professionals can interact.
- Stakeholders and their roles are important to understand. Pattern developed for RAVO are used here extensively to design a Business Model for N2SKY.

Table A.2: *RAVO: Query Interface*

Entity Name: Query Interface, Mandatory, 4-Service Layer		
Attributes /Modules	Description	Mandatory /Optional
Access_Mode()	Desktop, Mobile	M
Login()	Login to authenticate members of VO	O
Query_Processing()	Responsible for activities from query initiation to solution output	M
Query_Type()	Categorize according to the resource offered. An online expert opinion, download, resource request	M
Existing_Solution()	Searches the knowledge base of VO for existing solutions on the basis of parameter provided in the Query type. Successful search is return a problem solution. Unsuccessful search branches control to the VO management for finding a new solution from the scratch.	O
New_Solution()	It finds solution of the proposed problem (if Existing_Solution() is unsuccessful). User is provided with the appropriate output according the query	M
Response_Time()	Urgent/Normal, the user must be provided with a time frame depending upon the query type.	M
Input_Data()	Query string, necessary parameters	M
Output_Result()	Give back results to user. It could be notification as an email, a document, or a link to the Web site where results can be found. Resource access permission, unsuccessful processing status, contact information of an expert, Or any other method agreed upon by the participating entities	M

Table A.3: *N2SKY: Interface specification of the Query Interface component*

Query Interface			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	knowledgeBaseURL	URL,	*
	registryURL	URL	*
Inbound interface		Web	*
searchProblem()		synchr.	*
IN	problemName	String	*
OUT	proposedParadigms	nnDescription[]	*
searchSolution()		synchr.	*
IN	paradigmName	String	*
OUT	paradigm	nnDescription[]	*
Used components:	Knowledge base		

Table A.4: *RAVO: Domain Specific Application*

Entity Name: Domain Specific Application, Mandatory, 4-Service Layer		
Attributes /Modules	Description	Mandatory /Optional
Application_ID	Unique Application ID	M
Type	Standalone software, Online executable only,	M
Access_rights	Who can access this application	M
Application_Details()	Name, version, owner, volume, PC/mobile application, compatibility (OS support, memory etc)	M
Status	Free ware, trial, open source, paid	M

Table A.5: *N2SKY: Interface specification of the Web portal*

Web Portal			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	schemaLocation	URL, directory	*
	registryURL	URL	*
Inbound interface		Web	*
createNeuralObject()		synchr.	
...			
train()		asynchr.	*
...			
duplicateObject(()		synchr.	
...			
evaluate()		asynchr..	*
...			
showStatus()		synchr.	*
...			
Used components:	Registry		*
	Simulation Service		*
	Workflow System		
	Query interface		
	Annotation Interlayer		

Table A.6: *N2SKY: Interface specification of the Smartphone app*

Smartphone App			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	schemaLocation	URL, directory	
	registryURL	URL	*
Inbound interface		Smartphone UI	*
Operations see table A.5 (Web Portal)			
Used components:	Registry		*
	Simulation Service		*
	Workflow System		
	Query interface		
	Annotation Interlayer		

Table A.7: *N2SKY: Interface specification of the Hosted UI component*

Hosted UI			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		Web	*
Hosted UI functionality			
...			
Used components:	Hosted Component		*

Table A.8: *RAVO: Data Mining Tools*

Entity Name:Data Mining Tools, Optional, 4-Service Layer		
Attributes /Modules	Description	Mandatory /Optional
Tool_ID	Unique Tool ID	M
Purpose	Details of how this tool works and for what purpose	M
Access_rights	Who can access this application	M
Tool_Details()	Name, Version, Owner,	M
Manul()	A guide or instruction set for user explaining how it can be used efficiently	M

Table A.9: *RAVO: VO Trust*

VO Trust, Mandatory, 3-VO Layer		
Attributes /Modules	Description	Mandatory /Optional
Authentication()	Authenticates user as a VO member	M
Authorization()	Verifies the access rights assigned to the member according to a given role	M
Contract()	Contains sub modules i.e. Policy(), Goal(), Role(), Feedback()	M
Business_Model()	Contains sub modules Rules(), Roles(), Pricing_Algorithm(), Goal()	O
User_FeedBack()	Feed back from stakeholder is utilized to enhance the contract or Business Model. Change In requirement must be incorporated in contract or Business Model to keep the VO updated and evolve them dynamically	M
Resource_Infromation()	Resource Management() and Resource Catalogue()	

Table A.10: *N2SKY: Interface specification of the Management service component*

Management Service			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		Web Service	*
register_User()		synchr.	
...			
integrate_Paradigm()		asynchr.	*
...			
(integrate_Hardware()		synchr.	
...			
(buy_Package)		synchr..	*
...			
debit_Account()		synchr.	*
...			
credit_Account()		synchr.	*
...			
create_End_User_bill()		asynchr.	*
...			
check_Accounts()		asynchr..	*
...			
Used components:	Registry		*
	Controlling and Accounting		*
	Usermanagement		
	...		

Table A.11: *N2SKY: Interface specification of the Hosted Components*

Hosted Components			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		WS, API	*
Hosted Component functionality ...			
Used components:	Component Hosting Platform DBMS		*

Table A.12: *RAVO: User Interface*

Entity Name: User Interface, Mandatory, 3-VO Layer		
Attributes /Modules	Description	Mandatory /Optional
Login()	Authentication and Authorization	M
Query_Management()	Taking input parameters, processing query, displaying results processing query, displaying results taking feedback from user	M

Registry			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	schemaLocation	URL, directory	*
	registryURL	URL	*
	searchableAttributes	String , sep.	*
	holdbackTime	Long (sec)	*
refreshTime < holdbackTime	refreshTime	Long (sec)	*
Trigger	currentTime = lastRefresh + refreshTime		
checkRegisteredServices()	removeTime = currentTime + holdbackTime		
Inbound interface		WS	*
getHoldbackTime()		synchr.	*
	holdbackTime	Long (sec)	*
getSchemaLocations()		syn.	*
OUT	schemaLocations	String[]	*
getAllServiceIDs()		synchr.	*
OUT	serviceIDs	Long[]	*
getAllNNServiceIDs()		synchr.	Neur.
OUT	NNserviceIDs	Long[]	Neur.
submitServiceDescription()		synchr.	*
IN	serviceID or name	Long/String	*
OUT	ServiceDescription	String	*
submitNNServiceDescription()		synchr.	Neur.
IN	serviceID or name	Long, String	Neur.
OUT	nnDescription	nnDescription	Neur.
getServiceIDsByParadigm()		synchr.	Neur.
IN	paradigm	String	Neur.
OUT	serviceIDs	Long[]	Neur.
getSearchCriteria()		synchr.	
OUT	searchCriteria	String[]	
searchServicesByCriteria()		synchr.	
IN	searchCriteria	String[]	
OUT	serviceIDs	Long[]	
getTrainingEndpoint()		synchr.	Neur.
IN	serviceID	Long	Neur.
OUT	serviceEndpoint	URL	Neur.
getEvaluationEndpoint()		synchr.	Neur.
IN	serviceID	Long	Neur.
OUT	serviceEndpoint	URL	Neur.
registerService()		synchr.	*
IN	serviceName	String	*
IN	serviceDescription	String	*
IN	serviceEndpoint	URL	*
IN	searchCriteria	String[]	*
OUT	serviceID	Long	*
removeFromRegistry()		synchr.	*
IN	serviceID	Long	*
Used components:	-		

Figure A.3: *Interface specification of the Registry Component*

Table A.13: *RAVO: Resource Management*

Entity Name: Resource Management, Mandatory, 2- Abstract Layer		
Attributes /Modules	Description	Mandatory /Optional
Resource_ID	Unique resource identification	M
Resource_Type	Logical/Physical	M
Availability_Status	Resource is active part of VO	M
Resource_Provider _Information()	Resource provider information is maintained	M
Resource_Cost()	Resource usage policies or Business Model, which maintains resource cost and usage. Free in case of non-profit VO	M
Access_Rights()	Defined in Contract/Business Model	M
Resource_Scheduling()	How resources are aggregated for a problem solving activity.Different methods and algorithms are developed for this purpose	M
Resource_Consumption()	Percentage of the resources consumed in a problem solving activity	O
Resource_History()	Early participation in a problem solving activity and performance	O
Resource_Maintainance()	Add_resource(),Update_Resource(), Remove_Resource()	M

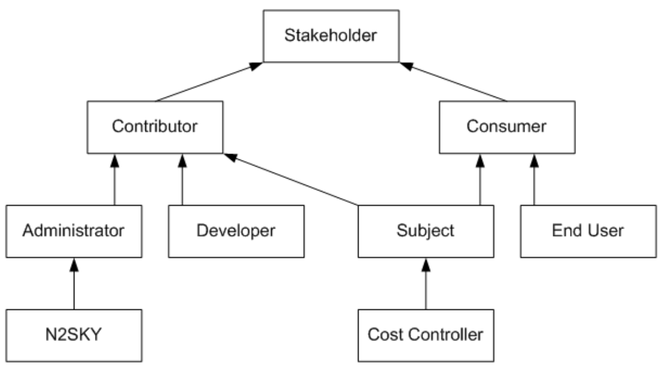

Figure A.4: *The Stakeholder Hierarchy in N2SKY*

Table A.14: *N2SKY: Interface specification of the SLA component*

SLA			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		Web Service	*
createSLAContract()		synchr.	*
IN	userID	long	*
IN	packageID	long	*
IN	expiry	Date/Time	*
IN	SLALevel	String	*
OUT	SLAContract	SLAContract	*
getSLAContract()		synchr..	*
IN	contractID	long	*
OUT	SLAContract	SLAContract	*
searchSLAContracts()		synchr.	
IN	searchParams	String[]	
OUT	SLAContracts	SLAContract[]	
Used components:	Registry		*
	DBMS		

Table A.15: *N2SKY: Interface specification of the User management component*

Controlling and Accounting			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		WS or API	*
buyPackage()		synchr.	*
...			
bookSingleUseCosts()		asynchr.	*
...			
calculateInvoice()		asynchr.	*
...			
bookPayment()		synchr.	*
...			
checkPayments()		asynchr.	*
...			
Used components:	Registry		*
	Access Control		*
	DBMS		

Table A.16: *N2SKY: Interface specification of the User management component*

Usermanagement			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		WS or API	*
CRUD User()		synchr.	*
...			
CRUD Role()		asynchr.	*
...			
Used components:	Registry		*
	Access Control		*
	DBMS		

Table A.17: *N2SKY: Interface specification of the Access Control component*

Access Control			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		WS or API	*
CRUD Resource()		synchr.	*
...			
CRUD Privilege()		synchr..	*
IN	resourceID	long	*
IN	userID	long	*
IN	expirationDate	DateTime	
OUT	privilegeID	long	*
...			
Used components:	Registry DBMS		*

Table A.18: *RAVO: Workflow Tools*

Entity Name: Workflow Mandatory, 2- Abstract Layer		
Attributes /Modules	Description	Mandatory /Optional
ID	Unique Workflow ID	M
Description	Sequential, state machine, data driven	M
Status	Start, end, proceeding, paused	M
Authorization_Information()	Who have right to access and call this module/Association with Roles	M
Interpretation_of_Workflow()	How Workflow provides information to the stakeholder /graphical, textual, source code, depending upon the mode it contacts other modules in the workflow management system to represent the information in an understandable form (code, markup languages, or a combination of both code and markup to author workflows.) Choice of approach depends on the authoring mode requirements for the solution.	M
Process_Management()	Includes Instance_Management() that controls the individual process instances to manage the concurrency	M

Table A.19: *N2SKY: Interface specification of the Workflow system component*

Workflow System			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		Web Service	*
Process Mgmt.		synchr.	*
...			
Instance Mgmt.		synchr./asynchr..	*
Controls the individual process instances to manage concurrency.			
Used components:	Registry		*
	Simulation Service		*
	...		

Table A.20: *RAVO: Graphical Interface*

Entity Name: Graphical Interface, Mandatory, 2- Abstract Layer		
Attributes /Modules	Description	Mandatory /Optional
GUI_ID	Unique ID	M
hline Input_Management()	Controls the input parameters for user interaction	
Processing_Management()	Controls the details (parameters) flowing among different modules	M
Output_Management()	Controls how results are displayed to the user and stored for the future use	M

Table A.21: *N2SKY: Interface specification of the Annotation Service component*

Annotation Service			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			
Inbound interface		WS or API	*
createAnnotation()		synchr.	*
IN	forObjectID	long	*
IN	text	String	*
IN	annotationType	String	*
IN	attachment	File	
OUT	annotationID as objectID		*
getAnnotations()		synchr.	*
IN	forObjectID	long	*
OUT	annotations	Annotation[]	*
editAnnotation()		synchr.	*
IN	annotationID	long	*
IN	changedText	String	*
OUT	changedText	String	*
deleteAnnotation()		synchr.	*
IN	annotationID	long	*
OUT	annotationID	long	*
Used components:	Knowledge Base		

Table A.22: *N2SKY: Interface specification of the Component Hosting Platform*

Component Hosting Platform			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		Web Service	*
getAvailableServer()		synchr.	*
startServer()		asynchr.	*
stopServer()		asynchr.	*
deploy()		asynchr.	*
undeploy()		asynchr.	*
getDeploymentURL()		synchr.	*
getAvailableDBMS()		synchr.	*
...			
Used components:	Registry		*
	Server		*
	DBMS		*
	Knowledge Base		

Table A.23: *RAVO: Infrastructure Enabler Layer*

Entity Name: Infrastructure_Enabler, Mandatory, 1-Infrastructure Enabler Layer		
Attributes /Modules	Description	Mandatory /Optional
QoS_Management()	Manages Quality of Service parameters agreed upon by participating organizations	M
SLA_Management()	Manages SLA agreed upon by participating organizations in Business Model or Contract	M
Security_Management()	Provides Security mechanism, secure communication and encryption facilities	M
Fault_Tolerance _Management()	Manages fault tolerance and disaster management, how to degrade gracefully instead of being crashed	M

Table A.24: *N2SKY: Interface specification of the Knowledge Management component*

Knowledge Management			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			
Inbound interface		WS, or API	*
CRUD operations over	**SPARQL queries**	synchr.	*
Used components:	Cloud Data Archive		

Table A.25: *RAVO: Resource Catalogue*

Entity Name: Resource Catalogue, Mandatory, 0-Factory Layer		
Attributes /Modules	Description	Mandatory /Optional
Category_ID	Identifies the category to which a resource belongs	M
Category_Type	hardware, Software, Logical	M
Status	Available or Not available	M
Resource_ID	Composite ID : Category_ID and Resource_ID	M
Resource_Type()	Computational, Storage, Data, Expert, Multimedia (Document, Audio, Video etc)	M
Access_Rights()	Defined according to the roles defined in Contract/Business Model	M
Add_Resource()	Resource Management	M
Remove_Resource()	Resource Management	M
Update_Resource()	Resource Management	M
Resource_Provider _Information()	Detailed information about the resource provider. Accessed via Resource_Provider_ID	M
Usage_Policy()	Details usage details and calculates cost for resource consumption. Legal terms and conditions associated with Resource. Resource provider also maintain these details for record.	M

Table A.26: *RAVO: Expert*

Entity Name:Expert Mandatory 0-Factory Layer		
Attributes /Modules	Description	Mandatory /Optional
Category_ID	Identifies the category to which a resource belongs	M
Category_Type	Logical	M
Resource_ID	Unique resource ID	M
Expert_Profile()	Details about expertise, domain, association/affiliations	M
Contact()	Email, Phone, Fax, timings of availability for online assistance	M
Availability_Status	Online/offline	M
affiliation	Individual or with en enterprize	M
Role_Assigned()	Stakeholder role (Subject/consumer/producer/administrator)	M
Resource_Provider_ID	In case of expert belonging to a participating organization	M

Table A.27: *RAVO: Data Services*

Entity Name: Data Service, Mandatory, 0-Factory Layer		
Attributes /Modules	Description	Mandatory /Optional
Category_ID	Identifies the category to which a resource belongs	M
Category_Type	Physical	M
Resource_ID	Unique resource ID	M
Availability_Status	Up/Down (resource is working correctly or not)	M
Resource_Cost()	Usage cost of the Data service	M
Access_Rights()	Authorization for utilizing Data service according to the Role assigned	M
Resource_Provider_ID	Unique ID	M

Table A.28: *RAVO: Computational Services*

Entity Name: Computational Services, Mandatory, 0-Factory Layer		
Attributes /Modules	Description	Mandatory /Optional
Category_ID	Identifies the category to which a resource belongs	M
Category_Type	Physical	M
Resource_ID	Unique resource ID	M
Resource_Provider_ID	Unique ID	M

Table A.29: *N2SKY: Interface specification of the Component Archive*

Component Replication Service			
Interface, operation, I/O	parameter name	param. type	mand.
Configuration			*
	registryURL	URL	*
Inbound interface		WS, API	*
archiveNewComponent() ...		asynchr.	*
copyComponentToNodes() ...		asynchr.	*
getAllArchivedComponents() ...		synchr	*
getArchivedComponentsOnNode() ...		synchr.	*
hasReplica() OUT:	 hasReplica	synchr. boolean	* *
deleteComponentFromNode() ...		asynchr.	*
Used components:	Registry Cloud Infrastructure		* *

Table A.30: *N2SKY: Interface specification of the Cloud Data Archive*

Data Archive		
Interface, service, description	type	mand.
Configuration		
Inbound interface	WS, API	*
Data request execution service (DRES) Is used to submit workflows, create sessions and get the status of synchronous requests.	synchr.	*
Data resource information service (DRIS) Is used to query information about a stored resource.	synchr.	*
Data sink service Is used to push data to data sinks.	synchr./asynchr.	*
Data source service Is used to pull data from data sources.	synchr./asynchr.	*
Session management service Is used to manage the lifetime of sessions.	synchr.	*
Request management service Is used to query request execution status subsequently of asynchronous requests.	synchr.	*
Used components:	Filesystem DBMS	*

Table A.31: *N2SKY: User Permission*

User Permission				
Permission to	N2SKY Contr.(NC)	Cost Contr. (CC)	Developer	End User
+/- NC Role	x	-	-	-
Reset Password	x	within Unit	-	-
+/- CC Role	x	within Unit	-	-
+/- Devel Role	x	within Unit	within Unit	-
+/- End User R	x	within Unit	within Unit	within Unit
Set Budget Limit	CC approvement	x	CC appr	CC appr

B Statement of Thesis

1. *“Virtual Organizations is an older concept which renews with the birth of collaborating computing paradigm. ”*
2. *“ Subject is a new concept, which presents the overlapped role of Resource, Producer and a Consumer.”*
3. *“ Virtual Organization can be defined in terms of Subject as - a set of cooperating building blocks, called Subjects.”*
4. *“ Categorization of resources into logical and physical, is a new dimension for VO developers. Inclusion of human expertise as a resource supports the demanding nature of problem solving ability, thereby increasing the level of trust in users.”*
5. *“RAVO is presented as a standard for formal and informal e-Collaborations in all domains.”*
6. *“ RAVO claims to be technology independent and flexible for future extensions.”*
7. *“ RAVO supports creations of VO both from scratch or evolution of an existing system. ”*
8. *“ Belief and Knowledge are two different entities. Sometimes it is hard to believe what we know and vice versa.”*
9. *“Hopes are ropes. Use them wisely to climb the targets, not to hang yourself and others.”*
10. *“Vienna is still a beautiful place to live, though I cannot speak Deutsche.”*

C Publications

Peer-Reviewed Papers

During my Ph.D studies, following research papers were published.

- W. Khalil, E. Schikuta. "Towards a Virtual Organization for Computational Intelligence". The Fourth International Conference on Digital Society, ICDS 2010, February 10-16, 2010 - St. Maarten, Netherlands Antilles.

  This paper presented conceptual basis for creating a VO from scratch. Patterns, Component identification and Gap analysis done for N2Grid in this paper is further generalized as Requirement Analysis Phase of RAVO. These patterns are presented in Chapter 3 of the thesis. *Subject* a new notion for the user in VO is the shift from the traditional role, which is also explained in detail in Chapter 4.

- W. Khalil, M. Juergen , E. Schikuta. "VELOCI: A Virtual E-learning Organization for Computational Intelligence". World Conference on Educational Multimedia, Hypermedia & Telecommunications ED-MEDIA 2010. June 29 - July 2, 2010- Toronto, Canada.

  This paper illustrated a novel, pure service-oriented E-learning system following the galaxy of services vision as a special case of generic Virtual Organization for Computational Intelligence. It implements collaboration of physical and logical resources in the shape of an integrated system. Chapter 5 includes the domain specific concrete Models for E-learning systems based on RAVO, presented in this paper.

- W. Khalil, J. Mangler and E. Schikuta. "Virtual Organization for Computational Intelligence (VOCI): Architecture and Realization". In Proceedings of the International Joint Conference on Neural Networks 2010 (WCCI2010), Barcelona, Spain, 2010.

  This paper presented the evolution of an existing system N2Grid to a VO for CI. The requitement analysis phase and generic architecture of RAVO are presented as a domain specific concrete model in the context of CI. *Subject* is further enhanced in the context of VO for CI. These concepts are presented in Chapter 5.

- W. Khalil and E. Schikuta. "Students and Teachers as Stakeholders in Virtual Organization based E-learning Systems". Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011 (pp. 1755-1761). ED-MEDIA 2011. Lisbon, Portugal

This paper presented a novel approach towards stakeholders in virtual organization based E-learning environment. Stakeholders were identified from the RAVO point of view. This pattern was extended to the domain of E-learning, where role of user was viewed as Subject (both consumer and contributor). Subject also act as a resource in the environment to pass on the knowledge as an expert of the field or by developing a new algorithm. Skills are viewed as a shared resource in the environment. Stakeholders were reviewed in the context of Virtual E-learning Organization for Computational Intelligence (VELOCI). Related research efforts were also presented. Discussion established a new design approach for the Virtual Organization based collaborative systems in general and for E-learning domain in specific. User is considered the center of design, part of the system rather than an external viewer of the entire system. Shortcomings in the existing approach were justified by proposing the novel view of Virtual Organization based E-learning system. These concepts are included in Chapter 4 and Chapter 5 of the thesis.

- W. Khalil and E. Schikuta. "Informal virtual organizations: A perfect home for subjects as building blocks". In ICDS 2011, The Fifth International Conference on Digital Society, (pp. 134-139). ICDS 2011. (Best Paper Award)

  This paper presented the concept of resources and users in both formal VOs and informal VOs. A resource hierarchy is defined and the role of a user as a resource was observed and discussed in different environments. The understanding of user roles is necessary for building a trust model for VOs. This approach was extended by a generic pattern for users in VOs and was justified using online social networks (e.g., Facebook). The concepts are elaborated with examples to understand when a user changes her role from a consumer to a resource and starts contributing to the environment. These research findings are included in Chapter 4 and Chapter 5 of the thesis.

Book Chapter

- Chapter title, "Virtual Organization for Computational Intelligence (VOCI)": Springer's new publishing project entitled Human-Computer Interaction: The Agency Perspective. Series: Studies in Computational Intelligence, Vol. 396, ISBN 978-3-642-25690-5.

  The research findings presented in this Book Chapter are incorporated in Chapter 5 of the thesis.

D Curriculum Vitae

Personal Data

First Name:	Wajeeha
Last Name:	Khalil
Nationality:	Pakistan
Gender:	Female
Date-of-Birth:	18-0-1980
Languages:	Urdu, English Pashto, Punjabi
Current Address:	Donaufelderstrasse 54/2121, 1210 Wien, Austria
Permanent Address:	H.No 1008,Haq Bahoo Street, Gulberg No 3, Peshawar Cantt, Pakistan
E-mail:	wk_ rehman1@yahoo.com

Education	Ph.D (Computer Science), University of Vienna, Austria (Pursuing since Mar.2008)
	MSc (Computer Science), University of Peshawar, Peshawar - 2003
	BSc (Computer Science), Jinnah College for Women, University of Peshawar - 2000
	(Pre-engineering) from F.G Degree College for Women, Peshawar - 1998
	Matric (Science) from F.G Girls High School, Peshawar - 1996

Research Experience

Currently, pursuing research in the area of SOA, Cloud based computing platforms and Virtual Organizations. My Ph.D research focuses on developing standard for creating Virtual Organization in form of Reference Architecture. During my Ph.D studies (2008-2012), following research papers were published.

- W. Khalil, E. Schikuta. "Towards a Virtual Organization for Computational Intelligence". The Fourth International Conference on Digital Society, ICDS 2010, February 10-16, 2010 - St. Maarten, Netherlands Antilles.

- W. Khalil, M. Juergen , E. Schikuta. “VELOCI: A Virtual E-learning Organization for Computational Intelligence”. World Conference on Educational Multimedia, Hypermedia & Telecommunications ED-MEDIA 2010. June 29 - July 2, 2010- Toronto, Canada.
- W. Khalil, J. Mangler and E. Schikuta. “Virtual Organization for Computational Intelligence (VOCI): Architecture and Realization”. In Proceedings of the International Joint Conference on Neural Networks 2010 (WCCI2010), Barcelona, Spain, 2010.
- W. Khalil and E. Schikuta. “Students and Teachers as Stakeholders in Virtual Organization based E-learning Systems”. Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011 (pp. 1755-1761). ED-MEDIA 2011. Lisbon, Portugal
- W. Khalil and E. Schikuta. “Informal virtual organizations: A perfect home for subjects as building blocks”. In ICDS 2011, The Fifth International Conference on Digital Society, (pp. 134-139). ICDS 2011. (Best Paper Award)

Book Chapter

- Chapter title, “Virtual Organization for Computational Intelligence (VOCI)”: Springer’s new publishing project entitled Human-Computer Interaction: The Agency Perspective. Series: Studies in Computational Intelligence, Vol. 396, ISBN 978-3-642-25690-5.

Research Activities in Pakistan

I joined research activities since October 2004. I have worked with a research group working in area of Grid Computing. During the research activities I co authored a paper on trust based resource selection in the area of Grid Computing (details followed here).

- Umar Farooq, Saeed Mahfooz and Wajeeha Khalil. “An Efficient Resource Prediction Model for Mobile Grids”, in the proceedings of PGNET2006, Liverpool JMU, Liverpool, 25-26 June, 2006.
- Umar Farooq and Wajeeha Khalil. “Grid as Humans Assistant; A logical solution provider to physical problems”, in the proceedings of CTS’06, Las Vegas, Nevada, USA, 14-17 May 2006. Published by IEEE Computer Society, accessible through IEEE & ACM Digital Libraries. ISBN No: 0-9785699-0-3

  Abstract: This paper presents theoretical foundations of a new vision regarding virtual organizations. The proposed vision exploits the concept of virtual organizations for the solution of human centered problems; mostly exist and solved in physical world in physical manner. The idea facilitates a user by providing complete logical solution (s) for an initiated problem. The idea pinpoints the need for the participation of different business organizations in a problem solution; as total requirements might not be available in a single organization or a user might not favor all resources of a single organization. This paper presents a contemporary view for the creation of virtual organization over grid infrastructure to assist humans in the solution of hard problems in terms of time, selection and integration. Proposed context of virtual organization believes in the construction of MAS environment. The proposed idea is equipped with a layered architecture for the construction of virtual organizations; a point missing in conventional MAS environments.

- Wajeeha Khalil. "Grids: Security Concerns", in PUTAJ, ISSN 1608-7925-Vol.13, 2006, pg 19.

  Abstract: Grid Computing is emerging as a new paradigm for next generation computing. It enables the sharing, selection and aggregation of geographically distributed heterogeneous resources for solving large-scale problems in science, engineering and commerce. Such information systems heavily rely upon the provision of adequate security. This paper reviews security issues related to grid computing. Various threats to Security are identified and the work done so far minimize these threats is presented in detail.

- Umar Farooq, M. Pasha, Wajeeha Rehman. "A contemporary vision of Virtual organization to Solve Human Centric Problems", in the proceedings of 3rd International Workshop on Frontiers of Information Technology, Islamabad, December 28, 29, 2005.

Distinction

- IARIA Best Paper Award

  Wajeeha Khalil and Erich Schikuta got the Best Paper Award of the International Academy, Research, and Industry Association at the ICDS 2011, Fifth International Conference on Digital Society, held in Gosier, Guadeloupe/France, February 23-28, 2011.

Scholarships

- HEC scholarship for Ph.D, in Vienna University of Technology, Vienna, Austria.

Work Experience

I have served the following institutions in different capacities,

- Visiting Faculty Member of Computer Science, University of Peshawar, Pakistan (2005-2006).
- Lecturer at Jinnah College for Women, University of Peshawar, Pakistan (2003-2008).

Glossary

A

A Reference Model for Collaborative Networks (ARCON)

A Reference Model for Collaborative Networks [8].

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon Elastic Compute Cloud[1] (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Application Programming Interface (API)

An application programming interface (API) is a set of functions that the operating system makes available to application programs for communicating with the other operating system [130].

Architecture

The organizational structure of a system.The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution [62].

B

Biomedical Informatics Research Network (BIRN)

The Biomedical Informatics Research Network[2] (BIRN) is a national initiative to advance biomedical research through data sharing and online collaboration. Funded by the National Institute of General Medicine Sciences (NIGMS), a component of the US National Institutes of Health (NIH), BIRN provides data-sharing infrastructure, software tools, strategies and advisory services, all from a single source[3].

[1] http://aws.amazon.com/ec2/
[2] http://www.birncommunity.org/
[3] http://www.birncommunity.org/

Blackboard

A platform for E-learning[4].

Business Architecture Project(BAP)

The central aim of the Business Architect Project (BAP) is to develop, implement and test management services and software tools which will facilitate the optimal design of virtual enterprises, enabling them to realize the value of business innovation[5](FEB 2000-JUL 2002) [24].

Business Integrator Dynamic Support Agents for Virtual Enterprize (BIDSAVER)

The BIDSAVER project aims at facilitating the constitution and the management of Virtual Enterprize and supporting their dynamic evolving configurations, driven by competitiveness-oriented criteria [131] [24]. FP5 IST (European Commission) (JAN 2001-JUN 2002)

C

caBIG

Cancer Biomedical Informatics Grid[6] (caBIG) creates a virtual network of interconnected data, individuals, and organizations that work together to redefine how cancer research is conducted.

Caroline

An advanced R&D project at Sun Microsystems, Project Caroline[7] is a hosting platform for development and delivery of dynamically scalable Internet-based services. It is designed to serve an emerging market of small and medium sized software-as-a-service (SaaS) providers.

Cloud Computing

A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [68].

[4] http://www.blackboard.com/
[5] http://www.uninova.pt/~cove/bk_profile.htm
[6] https://cabig.nci.nih.gov/
[7] http://labs.oracle.com/projects/caroline/

Cluster Computing

A cluster is a type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone computers working together as a single, integrated computing resource [132].

Collaborative Networked Organization (CNO)

A Collaborative Network (CN) is a network consisting of a variety of entities (e.g. organizations, people, machines) that are largely autonomous, geographically distributed, and hetereogenous in terms of their operating environment, culture, social capital and goals, but that collaborate to better achieve common or compatible goals, thus jointly generating value, and whose interactions are supported by computer networks. Most forms of collaborative networks imply some kind of organization over the activities of their constituents, identifying roles for the participants, and some governance rules, and therefore, can be called manifestations of Collaborative Networked Organizations (CNOs) [133].

Computational Intelligence and Machine Learning (CIML)

Computational Intelligence and Machine Learning (CIML), a virtual community for providing resources to researchers, students, and general public in the area of CI[8].

Computational Intelligence (CI)

A methodology involving computing that exhibits an ability to learn and/or to deal with new situations, such that the system is perceived to possess one or more attributes of reason, such as generalization, discovery, association and abstraction [134].

Computational Science (CS)

A multidisciplinary field which fuses three distinct interdisciplinary problem solving elements: algorithms and modeling and simulation software, computer and information science, and computing infrastructure [129].

Cooperative Environment Web Service (CEWebS)

CEWebS[9] is basically a Web-Service (SOAP) aggregator, that allows to subscribe to learning modules that are distributed throughout an organization. CEWebS is developed at the Department of Knowledge and Business Engineering and the Research Lab for Educational Technologies at the University of Vienna [120].

[8] http://www.cimlcommunity.org/
[9] http://www.cewebs.org/

Consumer

An entity in RAVO which interacts with a VO and consumes resources.

D

Data Grid

DataGrid[10] is a project funded by European Union. The objective is to build the next generation computing infrastructure providing intensive computation and analysis of shared large-scale databases, from hundreds of TeraBytes to PetaBytes, across widely distributed scientific communities.

Developer

In the context of RAVO, developer represent a class of participants who are professionals and responsible for developing domain specific application and framework supporting tools for VO.

Django

Django[11] is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.

E

e-COGON

e-COGNOS project aims to specify, develop, and deploy an innovative open model-based infrastructure and a set of tools that promote effective and consistent KM (including capturing, packaging, disseminating and reusing) within collaborative construction environments [131] [135].

E-collaboration

Alternative term for VO.

E-COLLEGE

Advanced Infrastructure for Pan-European Collaborative Engineering[12]. The goal

[10] http://www.twgrid.org/
[11] https://www.djangoproject.com/
[12] http://cic.vtt.fi/projects/voster/public.html

of the E-Colleg[13] project was to provide a new paradigm platform for distributed collaborative engineering through the definition and implementation of an advanced infrastructure for collaborative engineering [131]. FP5 IST (European Commission) (JAN 2000-DEC 2003)

E-learning

Electronic learning (E-learning) is instructional content or learning experience delivered or enabled by electronic technologies [121].

e-LEGAL

The goal of eLEGAL is to define a framework for legal conditions and contracts regarding the use of ICT in project business. The project will specify user requirements, implement legal support tools and promote an enhanced business practice in which the use of ICT in inter-enterprise information exchange is contractually stipulated[14].

e-MMEDIATE

Electronic Managing of Product Manufacturing, Engineering, Design and Investment Applying Information Technology for SMEs project developed a methodology for setting up and moving VE, especially consisting of small and medium sized enterprises SME, including the selection, implementation and usage of supporting IT [131].

e-Research E-Research or eResearch is a broad term used to describe a set of activities that harness the power of advanced information and communication technologies (ICTs) for research[15].

e-Science

e-Science[16] refer to the large scale science that will increasingly be carried out through distributed global collaborations enabled by the Internet. Typically, a feature of such collaborative scientific enterprises is that they will require access to very large data collections, very large scale computing resources and high performance visualisation back to the individual user scientists.

ECOLEAD[17]

European Collaborative networked Organisations LEADership initiative, ECOLEAD,

[13] http://www.ecolleg.org/
[14] http://cic.vtt.fi/projects/elegal/public.html
[15] http://www.eresearchsa.edu.au/whatis
[16] http://www.nesc.ac.uk/nesc/define.html
[17] http://www.ve-forum.org

aims to create strong foundations and mechanisms needed to establish the most advanced collaborative and network-based industry society in Europe: "In ten years most enterprises will be part of some sustainable collaborative networks that will act as breeding environments for the formation of dynamic virtual organizations in response to fast changing market conditions.

EGEE

Enabling Grid for E-sciencE[18] (EGEE) project is to build on recent advances in grid technology and develop a grid service infrastructure in European which is available to scientists 24 hours-a-day.

Everything as a Service (XaaS)

A popular Cloud service model which means that products and services are to be released, sold, bought and used as "services" [136].

Extended Enterprize Resources, Network Architectures and Learning (EXTERNAL)

EXTERNAL[19] addresses the challenges met when forming an extended enterprize (EE), characterized by a dynamic and time-limited collaboration between business partners.The goal of EXTERNAL is to provide solutions that make this collaboration effective and repeatable. The objectives of EXTERNAL are focusing at developing methodology, infrastructure/tools and business solutions for EE modelling, analysis, engineering and operation. Also process learning; deployment of open knowledge-sharing infrastructures and validation of results will be tasks of the project.

F

Force.com

Force.com[20] is the proven cloud platform to automate and extend your business and deliver the social enterprize.

Framework

A set of assumptions, concepts, values, and practices that constitutes a way of viewing the current environment [62].

[18] http://www.twgrid.org/
[19] http://cordis.europa.eu/fetch?CALLER=PROJ_ICT_TEMP&ACTION=D&CAT=PROJ&RCN=54343
[20] http://www.force.com/

G

GEON

GEON[21] is an open collaborative project that is developing cyberinfrastructure for integration of 3 and 4 dimensional earth science data [59].

Grid Computing

A grid enables the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organizations for solving large-scale resource intensive problems in science, engineering, and commerce [83].

H

High Performance Computing (HPC)

A High Performance Computer (HPC) is usually defined as computer hardware based on vector or multi processor parallel computers (or some mixture) that offers atleast a two orders of magnitude increase in computing power than is available from a top-end workstation.

I

ICCI

Innovation co-ordination, transfer and deployment through networked Co-operation in the Construction Industry[22] (ICCI).

Information Communication Technology (ICT)

ICT (information and communications technology - or technologies) is an umbrella term that includes any communication device or application, encompassing: radio, television, cellular phones, computer and network hardware and software, satellite systems and so on, as well as the various services and applications associated with them, such as videoconferencing and distance learning. ICTs are often spoken of in a particular context, such as ICTs in education, health care, or libraries[23].

Infrastructure as a Service (IaaS)

[21] http://www.geongrid.org/index.php/about/
[22] http://cic.vtt.fi/projects/icci/public.html
[23] http://searchcio-midmarket.techtarget.com/definition/ICT

IaaS is sometimes considered to be the provision of computer infrastructure (typically a platform visualization environment) as a service [79].

Intellectual Property (IP)

Intellectual property [24] (IP) refers to creations of the mind: inventions, literary and artistic works, and symbols, names, images, and designs used in commerce.

ISTforCE

ISTforCE[25] is a European 5th Framework Information Society Technologies project. The acronym stands for Intelligent Services and Tools for Concurrent Engineering.

L

Large Hadron Collider (LHC)

The Large Hadron Collider[26] (LHC) is the world's largest and highest-energy particle accelerator. It was built by the European Organization for Nuclear Research (CERN) over a ten year period from 1998 to 2008, with the aim of allowing physicists to test the predictions of different theories of particle physics and high-energy physics, and particularly for the existence of the hypothesized Higgs boson and of the large family of new particles predicted by supersymmetry [58].

Linked Environments for Atmospheric Discovery (LEAD)

LEAD[27] is a VO for identifying, accessing, preparing, assimilating, predicting, managing, analyzing, mining, and visualizing a broad array of meteorological data and model output, independent of format and physical location.

M

Moodle Moodle[28]is a Course Management System (CMS), also known as a Learning Management System (LMS) or a Virtual Learning Environment (VLE). It is a Free web application that educators can use to create effective online learning sites.

myExperiment

[24] http://www.wipo.int/about-ip/en/
[25] http://istforce.eu-project.info/
[26] http://lhc.web.cern.ch/lhc/
[27] http://portal.leadproject.org/gridsphere/gridsphere
[28] http://moodle.org/

myExperiment[29] is a collaborative environment where scientists can safely publish their workflows and experiment plans, share them with groups and find those of others. Workflows, other digital objects and bundles (called Packs) can now be swapped, sorted and searched like photos and videos on the Web. Unlike Facebook[30] or MySpace[31], myExperiment[32] fully understands the needs of the researcher and makes it really easy for the next generation of scientists to contribute to a pool of scientific methods, build communities and form relationships - reducing time-to-experiment, sharing expertise and avoiding reinvention. myExperiment is now the largest public repository of scientific workflows and is Linked Data compliant.

N

N2Cloud

N2Cloud [96], a novel cloud-based NN simulation system, which provides and exchanges NN knowledge and simulation resources to and between arbitrary users on a world-wide basis following the Web 2.0 principle. N2Cloud enables the exchange of knowledge, as NN objects and paradigms, by a VO environment and delivers ample resources by exploiting the cloud computing principle.

N2Grid

N2Grid [94] [108] is a system for the usage of NN resources on a world-wide basis. The approach employs the infrastructure of the grid as a transparent environment to allow users the exchange of information (NN resources, as NN objects and NN paradigms) and exploit the available computing resources for NN specific tasks leading to a grid based, world-wide distributed, NN knowledge and simulation system.

N2Sky

N2Sky is a cloud based VO for NN under development at University of Vienna. It being developed as a realization of our proposed framework RAVO. Details are available in Chapter 6 and Appendix A.

nanoHUB

nanoHUB.org[33] The nanoHUB is an online portal for nanotechnology researchers, instructors, and students created by Purdue University and the National Science

[29] http://wiki.myexperiment.org/index.php/Main_Page
[30] www.facebook.com
[31] www.myspace.com
[32] http://www.myexperiment.org/
[33] http://nanohub.org/topics/ElectronicsFromTheBottomUp

Foundation. It uses cyberinfrastructure to provide access to scientific tools for research, demonstration, and collaboration, as well as instructional material [137].

National Science Foundation (NSF)

The National Science Foundation[34] (NSF) is an independent federal agency created by Congress in 1950 "to promote the progress of science; to advance the national health, prosperity, and welfare; to secure the national defense.."

Nexof

The overall ambition of the NEXOF-RA[35] project is to build a Reference Architecture for the NESSI Open Framework (ranging from the infrastructure up to the interfaces with the end users) leveraging research in the area of service-based systems to consolidate and trigger innovation in service oriented economies.

O

Open Grid Forum (OGF)

Open Grid Forum[36] (OGF) is an open community committed to driving the rapid evolution and adoption of applied distributed computing. Applied Distributed Computing is critical to developing new, innovative and scalable applications and infrastructures that are essential to productivity in the enterprize and within the science community.

Open Science Grid (OSG)

The Open Science Grid[37] (OSG) advances science through open distributed computing. The OSG is a multi-disciplinary partnership to federate local, regional, community and national cyberinfrastructures to meet the needs of research and academic communities at all scales.

OpenID

OpenID[38] allows you to use an existing account to sign in to multiple websites,

[34] http://www.nsf.gov/index.jsp
[35] http://www.nexof-ra.eu/
[36] http://www.gridforum.org/
[37] http://www.opensciencegrid.org/
[38] http://openid.net/get-an-openid/what-is-openid/

without needing to create new passwords.

OSMOS

The OSMOS[39] (Open System for Inter-enterprize Information Management in Dynamic Virtual Environments) project is specifically concerned with defining the working practices, processes, techniques, tools and technical infrastructure to allow the European construction industry to progress from its current position towards a large scale, computer integrated approach. As such, it is an industry-led project involving construction end-users, construction IT providing companies, and academic and research organizations.

OASIS

OASIS[40] (Organization for the Advancement of Structured Information Standards) is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society.

P

Platform as a Service (PaaS)

PaaS is defined as the delivery of a computing platform and solution stack as a service. It often goes further with the provision of a software development platform that is designed for *cloud computing* at top of the cloud stack. It provides computational resources via a platform upon which applications and services can be developed and hosted (e.g. Force.com, Google App Engine, Windows Azure Platform) [16].

prodAEC

prodAEC[41] is a European Network for Product and Project Data Exchange, e-Work and e-Business in Architecture, Engineering and Construction Thematic Network under IST KA VIII.1.2, (2001-2003).

PRODCHAIN

Development of a decision support methodology to improve logistics performance of globally acting production networks[42].

R

[39] http://cic.vtt.fi/projects/osmos/main.html
[40] http://www.oasis-open.org/org
[41] http://cic.vtt.fi/projects/prodaec/index.html
[42] http://www.uninova.pt/~cove/projects.htm

Reference Architecture RA

An RA is defined as a way of documenting good architecture design practice to address commonly occurring problem [62].

Reference Architecture for Virtual Organization (RAVO)

Proposed standard for creation and managing the VO. Discussed in detail in Chapter 3.

Reference Model

A reference model is a division of functionality together with data flow between the artefacts [138].

S

Sakai

A vibrant community creating technology that enhances teaching, learning and research [43].

SAP Business ByDesign

SAP Business ByDesign[44] provides intuitive navigation, embedded analytics and built-in learning capabilities. It is easy to deploy, easy to adopt and doesn't require additional investment in IT infrastructure and staff.

SCOR Model

The Supply Chain Operations Reference[45] (SCOR) model is the product of Supply Chain Council (SCC), an independent, nonprofit, global corporation with membership open to all companies and organizations interested in applying and advancing the state-of-the-art in supply chain management systems and practices.

Service

A *service* is defined as a function that is well-defined, self-contained, and does not depend on the context or state of other services [63].

[43] http://sakaiproject.org/
[44] http://www.sap.com/solutions/products/sap-bydesign/what-is-sap-business-bydesign/overview/index.epx
[45] http://supply-chain.org/scor

Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) speaks of a collection of services, which communicate with each other, e.g., simple data passing or two or more services coordinating an activity. The goal of the SOA Reference Architecture is to provide a blueprint for creating or evaluating an architecture Additionally, it provides patterns and insights for integrating these fundamental elements of an SOA as exemplified in the layers of an SOA[46].

SHAMAN

SHAMAN[47] (Sustaining Heritage Access through Multivalent ArchiviNg) is a Large Integrated Project co-financed by the European Union within the Seventh Framework Programme. SHAMAN aims to create a technology environment which may be used to manage the storage, access, presentation, and manipulation of potentially any digital object over time.

Software as a Service (SaaS)

SaaS is defined as a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own Web servers or upload the application to the consumer device, disabling it after use or after the on-demand contract expires.

SolProv

The goal of this query interface, SolProv (Solution Provider), is to allow the user to specify her query in form of a natural language description of the problem statement [112].

SPI Stack

Cloud stack based on SaaS, PaaS, IaaS.

Southern California Earthquake Center (SCEC)

The Southern California Earthquake Center[48] (SCEC) is a community of over 600 scientists, students, and others at over 60 institutions worldwide, headquartered at the University of Southern California. SCEC is funded by the National Science Foundation and the U.S. Geological Survey to develop a comprehensive understanding of earthquakes in Southern California and elsewhere, and to communicate useful knowledge for reducing earthquake risk.

[46] http://www.opengroup.org/projects/soa-ref-arch/
[47] http://shaman-ip.eu/shaman/
[48] http://www.scec.org/

Stakeholder

Definition of stakeholder in VO environment is domain dependant. The IEEE Standard 1471-2000 [62] defines the stakeholder as

- The user of the system.
- Those responsible by the acquisition of the system.
- The developers and providers of the system's technology.
- The maintainers of the system as a technical operational entity.

Subject

A component of VO, which can consume the resources, offered and also can act like a resource to be consumed in the VO environment [1] [3] [2].

System Architecture Forum (SAF)

Discuss practices, research, and lessons learned with regard to the practical development, implementation and management of system architectures[49].

T

TeraGrid

TeraGrid[50] is a effort to build and deploy the world's largest, most comprehensive, distributed infrastructure for open scientific research.By 2004, the TeraGrid will include 20 teraflops of computing power distributed at nine sites, facilities capable of managing and storing nearly 1 petabyte of data, high-resolution visualization environments, and toolkits for grid computing. Currently, it is replaced by XSEDE.

V

View

View[51] is a representation or description of the entire system from a single perspective. Stakeholder is the viewer, who perceives the system according to her role.

[49] http://www.architectingforum.org/
[50] http://www.twgrid.org/
[51] http://shaman-ip.eu/(EuropeanCommission,ICT-216736)

Viewpoint

A pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis [97].

VELOCI

A proposed platform for CI community supporting E-learning (VELOCI) [2].

Virtual Enterprize

Alternative term for VO.

Virtual Environment (VE)

Virtual Environment are defined as interactive, virtual image displays enhanced by special processing and by nonvisual display modalities, such as auditory and haptic to convince users that they are immersed in a synthetic space [139]. It is alternatively called" Virtual Reality" or "Virtual World".

Virtual Machine (VM)

A virtual machine is implemented by adding a layer of software to a real machine to support the desired virtual machine's architecture [140].

Virtual Organization (VO)

VO is sharing the geographically dispersed resources for achieving a common goal. A VO can comprise a group of individuals whose members and resources may be dispersed geographically and institutionally, yet who function as a coherent unit through the use of cyber-infrastructure (CI) [10].

Virtual Organization for Computational Intelligence (VOCI)

A a proposed platform for CI community [3].

Virtual Team

A virtual team, like every team, is a group of people who interact through interdependent tasks guided by common purpose. Unlike conventional teams, a virtual team works across space, time, and organizational boundaries with links strengthened by webs of communication technologies [141].

Virtual Organization for Computational Science (VOCS)

A proposed platform for CS community. Defined in detail in Chapter 5, section 5.6.

VOSTER

Aim of VOSTER is to collect, analyZe and synthesize the results from a number of leading European research projects on Virtual Organization (VO), i.e. geographically distributed, functionally and culturally diverse, dynamic and agile organizational entities linked through ICT [52].

W

WebCT

WebCT[53] (Course Tools) or Blackboard Learning System, now owned by Blackboard, is an online proprietary virtual learning environment system that is sold to colleges and other institutions and used in many campuses for e-learning.

Windows Azure

Windows Azure[54] is an open and flexible cloud platform that enables user to quickly build, deploy and manage applications across a global network of Microsoft-managed data centers. User can build applications using any language, tool or framework. User can integrate her public cloud applications with her existing IT environment.

X

XSEDE

The Extreme Science and Engineering Discovery Environment[55] (XSEDE) is the most advanced, powerful, and robust collection of integrated advanced digital resources and services in the world. It is a single virtual system that scientists can use to interactively share computing resources, data, and expertise.

[52] http://cic.vtt.fi/projects/voster/public.html
[53] http://en.wikipedia.org/wiki/WebCT
[54] http://www.windowsazure.com/en-us/home/tour/overview/
[55] https://www.xsede.org/web/guest/overview

Bibliography

[1] Wajeeha Khalil and Erich Schikuta. Towards a virtual organisation for computational intelligence. 2010. Accepted in The Fourth International Conference on Digital Society ICDS 2010, February 10-16, 2010 - St. Maarten, Netherlands Antilles.

[2] Wajeeha Khalil, Juergen Mangler, and Erich Schikuta. VELOCI: A virtual e-learning organization for computational intelligence. In *World Conference on Educational Multimedia, Hypermedia; Telecommunications ED-MEDIA 2010*, Toronto, Canada, 6 2010.

[3] Wajeeha Khalil, Jurgen Mangler, and Erich Schikuta. Virtual organization for computational intelligence (VOCI):architecture and realization. In *International Joint Conference on Neural Networks 2010 (WCCI2010)*, Barcelona, Spain, 2 2010.

[4] Wajeeha Khalil and Erich Schikuta. Students and teachers as stakeholders in virtual organization based e-learning systems. In Theo Bastiaens and Martin Ebner, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2011*, pages 1755–1761, Lisbon, Portugal, June 2011. AACE.

[5] W. Khalil and E. Schikuta. Informal virtual organizations: A perfect home for subjects as building blocks. In *ICDS 2011, The Fifth International Conference on Digital Society*, pages 134–139, 2011.

[6] Erwin Mann. N2sky. A cloud based Neural Network Virtual Environment (Master Dissertation in process)., January 2012.

[7] Bernhard Katzy, Chunyan Zhang, and Herman Löh. Reference models for virtual organisations. In Luis M. Camarinha-Matos, Hamideh Afsarmanesh, and Martin Ollus, editors, *Virtual Organizations*, pages 45–58. Springer US, 2005.

[8] Weiming Shen, Luis Camarinha-Matos, and Hamideh Afsarmanesh. Towards a reference model for collaborative networked organizations. In *Information Technology For Balanced Manufacturing Systems*, volume 220 of *IFIP International Federation for Information Processing*, pages 193–202. Springer Boston, 2006.

[9] Peter F. Drucker. *Post-Capitalist Society.* HarperCollins, November 2009.

[10] Carl Kesselman, Ian Foster, Jonathon Cummings, Katherine A. Lawrence, and Thomas Finholt. Beyond being there: A blueprint for advancing the design, development, and evaluation of virtual organizations. Technical report, May 2008.

[11] W.A. Wulf. The collaboratory opportunity. *Science*, 261(5123):854–855, 1993.

[12] Tony Hey and Anne E. Trefethen. E-science, cyberinfrastructure and grid middleware services. *Science*, 308:817–812, 2005.

[13] Michael O'Leary and Jonathon Cummings. The spatial, temporal, and configurational characteristics of geographic dispersion in teams. *MIS Quarterly*, 31:433–452, March 2007.

[14] Preece Jennifer. *Online communities: Designing usability and supporting sociability*. John Wiley & Sons Ltd, Chichester, England, 2000.

[15] D.C. Plummer, T.J. Bittman, T. Austin, D.W. Cearley, and D.M. Smith. Cloud computing: Defining and describing an emerging phenomenon. *Gartner, June*, 17, 2008.

[16] K. Jeffery and B. Neidecker-Lutz. The future of cloud computing opportunities for european cloud computing beyond 2010. *Expert Group report, public version*, 1, 2010.

[17] H.B. Thorelli. Networks: between markets and hierarchies. *Strategic management journal*, 7(1):37–51, 1986.

[18] J.A. Byrne, R. Brandt, and O. Port. The virtual corporation. *Business week*, 8(1993):98–103, 1993.

[19] G Weber and F Walsh. The virtual organisation. The Gabler Magzine Number 6-7, 1994.

[20] B.R. Katzy. Design and implementation of virtual organizations. In *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, volume 4, pages 142–151.

[21] B. Travica. The design of the virtual organization: A research model. In *Proceedings of the Americas Conference on Information Systems. August*, pages 15–17, 1997.

[22] Y.P. Shao, M.K.O. Lee, and S.Y. Liao. Virtual organizations: the key dimensions. In *Research Challenges, 2000. Proceedings. Academia/Industry Working Conference on*, pages 3–8. IEEE, 2000.

[23] H. Jägers, W. Jansen, and W. Steenbakkers. Characteristics of virtual organizations. *Organizational virtualness*, page 65, 1998.

[24] A. Abuelmaatti and Y. Rezgui. Virtual organizations in practice: A european perspective. *AMCIS 2008 Proceedings*, page 142, 2008.

[25] David J. Skyrme. *Knowledge networking: creating the collaborative enterprise.* Butterworth-Heinemann, November 1999.

[26] David Skyrme Associates. I$\hat{3}$ update / entovation international news, no. 30. Accessible at : http://www.skyrme.com/updates/u30.htm, June 1999.

[27] A. Mowshowitz. Virtual organization. *Communications of the ACM*, 40(9):30–37, 1997.

[28] M. Kürümlüoglu, R. Nøstdal, and I. Karvonen. Base concepts. *Virtual Organizations*, pages 11–28, 2005.

[29] Y. Rezgui, I. Wilson, W. Olphert, and L. Damodaran. Socio-organizational issues. *Virtual Organizations*, pages 187–198, 2005.

[30] I. Zigurs. Leadership in virtual teams: Oxymoron or opportunity? *Organizational Dynamics*, 31(4):339–351, 2003.

[31] J. Lipnack and J. Stamps. *Virtual teams: People working across boundaries with technology*. John Wiley & Sons Inc, 2000.

[32] P. Barrett and M. Sexton. Innovation in small, project-based construction firms*. *British Journal of Management*, 17(4):331–346, 2006.

[33] K.S. Pawar and S. Sharifi. Virtual collocation of design teams: coordinating for speed. *International Journal of Agile Management Systems*, 2(2):104–113, 2000.

[34] M. Shelbourn, T. Hassan, and C. Carter. Legal and contractual framework for the vo. *Virtual Organizations*, pages 167–176, 2005.

[35] S.L. Jarvenpaa, K. Knoll, and D.E. Leidner. Is anybody out there?: antecedents of trust in global virtual teams. *Journal of Management Information Systems*, 14(4):29–64, 1998.

[36] Elena Rocco. Trust breaks down in electronic contexts but can be repaired by some initial face-to-face contact. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '98, pages 496–502, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[37] E. Rocco, T.A. Finholt, E.C. Hofer, and JD Herbsleb. Out of sight, short of trust. In *Presentation at the Founding Conference of the European Academy of Management. Barcelona, Spain*, 2001.

[38] Thomas Finholt and Jeremy Birnholtz. If we build it, will they come? the cultural challenges of cyberinfrastructure development. In WILLIAM BAINBRIDGE and MIHAIL ROCO, editors, *Managing nano-bio-info-cogno innovations*, pages 89–101. Springer Netherlands, 2006.

[39] W. Alsakini, J. Kiiras, and P. Huovinen. Competitive virtuality among construction management services company. *Encyclopaedia of Networked and Virtual Organizations, Goran D. Putnik, Maria M. Cunha, eds. Information Science Reference*, 2006.

[40] UJ Franke. Virtual web organizations & market conditions. *The electronic journal of organizational virtualiness*, 3(4):43–64, 2001.

[41] L. Arnison and P. Miller. Virtual teams: a virtue for the conventional team. *Journal of Workplace Learning*, 14(4):166–173, 2002.

[42] K. Helling, M. Blim, and B. O'Regan. An appraisal of virtual networks in the environmental sector. *Management of Environmental Quality: An International Journal*, 16(4):327–337, 2005.

[43] Y. Rezgui. Exploring virtual team-working effectiveness in the construction sector. *Interacting with Computers*, 19(1):96–112, 2007.

[44] T. Kayworth and D. Leidner. The global virtual manager: a prescription for success. *European Management Journal*, 18(2):183–194, 2000.

[45] E.F. McDonough III, K.B. Kahnb, and G. Barczaka. An investigation of the use of global, virtual, and colocated new product development teams. *Journal of Product Innovation Management*, 18(2):110–120, 2001.

[46] G.A. Yukl and G. Yukl. *Leadership in organizations*. Prentice Hall Upper Saddle River, NJ, 2002.

[47] D. Walters. Virtual organisations: new lamps for old. *Management Decision*, 38(6):420–436, 2000.

[48] Charlie Catlett, WE Allcock, P. Andrews, R. Aydt, R. Bair, N. Balac, B. Banister, T. Barker, M. Bartelt, P. Beckman, et al. Teragrid: Analysis of organization, system architecture, and middleware enabling new types of applications. *HPC and Grids in Action, Amsterdam*, 2007.

[49] Open science grid. Accessible at: http://www.opensciencegrid.org/.

[50] R. Bradt. Virtual organisations: a simple taxonomy, 1998.

[51] J. Burn and M. Barnett. Virtual cabbages-models for making kings in the e-grocery business. 2000.

[52] H.J. Harrington. *Business process improvement: The breakthrough strategy for total quality, productivity, and competitiveness*. McGraw-Hill Professional, 1991.

[53] Ian Foster and Carl Kesselman. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann, 2004.

[54] Accessible at : http://portal.leadproject.org/gridsphere/gridsphere.

[55] G. Klimeck, M. McLennan, M. Mannino, M. Korkusinski, C. Heitzinger, R. Kennell, and S. Clark. NEMO 3-D and nanoHUB: Bridging Research and Education. In *Nanotechnology, 2006. IEEE-NANO 2006. Sixth IEEE Conference on*, volume 2, pages 441–444. IEEE, 2006.

[56] H. Magistrale, S. Day, R.W. Clayton, and R. Graves. The scec southern california reference three-dimensional seismic velocity model version 2. *Bulletin of the Seismological Society of America*, 90(6B):S65, 2000.

[57] Andrew C. von Eschenbach and Kenneth Buetow. Cancer Informatics Vision: caBIG. *Cancer informatics*, 2:22, 2006.

[58] L.H. Collider. Large hadron collider. 2008. Accessible at: http://www.abitabout.com/Large+Hadron+Collider.

[59] U. Nambiar, B. Ludaescher, K. Lin, and C. Baru. The GEON portal: accelerating knowledge discovery in the geosciences. In *Proceedings of the 8th annual ACM international workshop on Web information and data management*, pages 83–90. ACM, 2006.

[60] Ian Foster. Service-oriented science: Scaling escience impact. In *IAT '06: Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, pages 9–10, Washington, DC, USA, 2006. IEEE Computer Society.

[61] Gerrit Muller. A Reference Architecture Primer. *Eindhoven Univ. of Techn., Eindhoven, White paper*, 2008. Accessible at: http://www.gaudisite.nl/referencearchitectureprimerslides.pdf.

[62] Rich Hilliard. IEEE-Std-1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems. *IEEE, http://standards. ieee.org*, 2000.

[63] Douglas K. Barry. *Web services and service-oriented architecture: the savvy manager's guide*. Morgan Kaufmann, April 2003.

[64] SOA reference architecture. Accessible at: http://www.opengroup.org/projects/soa-ref-arch/uploads/40/19713/soa-ra-public-050609.pdf, April 2009.

[65] A Malis. Routing over large clouds (rolc) charter. Part of the 32nd IETF meeting minutes, 1993.

[66] JOHN MARKOFF. Internet critic takes on microsoft. The Nework Times, April 2001. Accessible at: http://www.nytimes.com/2001/04/09/technology/09HAIL.htmlex =1230872400&en=5d156fc75d409335&ei=5070.

[67] Jeff Barr. Amazon ec2 beta. Accessible at: http://aws.typepad.com/aws/2006/08/amazon_ec2_beta.html, August 2006.

[68] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08*, pages 1–10. Ieee, 2008.

[69] M.A. Vouk. Cloud computing–issues, research and implementations. *Journal of Computing and Information Technology*, 16(4):235–246, 2004.

[70] Ibm introduces ready-to-use cloud computing collaboration services get clients started with cloud computing. Accessible at: http://www-03.ibm.com/press/us/en/pressrelease/22613.wss, October 2007.

[71] Geoffrey Raines. Cloud computing and soa. Technical report, The MITRE Corporation, November 2009.

[72] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm. What's inside the cloud? an architectural map of the cloud landscape. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pages 23–31. IEEE Computer Society, 2009.

[73] D. Amrhein, P. Anderson, and A. de Andrade. Cloud computing use cases. *A white paper produced by the Cloud Computing Use Case Discussion Group, Tech. Rep. Version*, 2, 2009.

[74] Bob Scheifler Richard Zippel, John McClain and Thomas Vinod Johnson. Project caroline: Platform as a service,sun microsystems, inc. Accessible at: http://labs.oracle.com/sunlabsday/docs.2008/caroline-labs-open-house.6.pdf, 2008.

[75] Django framework. Accessible at: https://www.djangoproject.com/.

[76] Google inc. google apps engine. Accessible at: http://code.google.com/intl/en/appengine/.

[77] Joyent. reasonably smart. Accessible at: http://www.joyent.com/.

[78] Microsoft. azure services platform. Accessible at: http://www.microsoft.com/windowsazure/.

[79] Nexof ra. Accessible at: http://www.nexof-ra.eu/.

[80] C. Kesselman and I. Foster. The grid: blueprint for a new computing infrastructure. 1998.

[81] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of High Performance Computing Applications*, 15(3):200, 2001.

[82] I. Foster. What is the grid? a three point checklist. *GRID today*, 1(6), 2002.

[83] M. Baker, R. Buyya, and D. Laforenza. Grids and grid technologies for wide-area distributed computing. *Software: Practice and Experience*, 32(15):1437–1466, 2002.

[84] Enabling grid for e-science portal. Accessible at: http://www.eu-egee.org/.

[85] Earth system grid center for enabling technologies (esg-cet).

[86] Ian Foster. Cloud, grid, what's in a name? Accessible at: http://ianfoster.typepad.com/blog/2008/08/cloud-grid-what.html, August 2008.

[87] An egee comparative study: Grids and clouds -evolution or revolution? Technical report, CERN, 2008.

[88] LM Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. A break in the clouds: towards a cloud definition, sigcomm comput. *Commun. Rev*, 39(1):50–55, 2009.

[89] Grid vs. cloud vs. what really matters. Accessible at: http://www.hpcinthecloud.com/hpccloud/2008-08-22/grid_vs_cloud_vs_what_really_matters.html, August 2008.

[90] Dinkar Sitaram and Geetha Manjunath. *Moving To The Cloud: Developing Apps in the New World of Cloud Computing*. Elsevier, December 2011.

[91] Lee Gillam. *Cloud Computing: Principles, Systems and Applications*. Springer, August 2010.

[92] J.A. Estefan, K. Laskey, F.G. McCabe, and D. Thornton. Reference architecture for service oriented architecture version 1.0. *Accessible at: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf*, 2008.

[93] Ensure: Enabling knowledge sustainability, usability and recovery for economic value, eu fp7 project,. Accessible at: http://cordis.europa.eu/fetch?CALLER=PROJ_ICT&ACTION =D&CAT=PROJ&RCN=98002.

[94] Erich Schikuta and Peter Paul Beran. A gridified artificial neural network resource. In *Tools with Artificial Intelligence. ICTAI 2007*, volume 2, pages 396–404, Oct 2007.

[95] Dennis Gannon. Building virtual organizations around super computing grids and clouds, January 2008.

[96] Altaf Ahmad Huqqani, Peter Beran, Xin Li, and Erich Schikuta. . N2cloud: Cloud based neural network simulation application. In *In Proceedings of the International Joint Conference on Neural Networks 2010 (WCCI 2010), Barcelona, Spain*, 2010.

[97] L. Jen and Y. Lee. Working group. ieee recommended practice for architectural description of software-intensive systems. In *IEEE Architecture*. Citeseer, 2000.

[98] S. Rinderle-Ma and M. Reichert. Managing the life cycle of access rules in ceosis. In *Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE*, pages 257–266. IEEE, 2008.

[99] S. Rinderle, M. Reichert, and P. Dadam. Correctness criteria for dynamic changes in workflow systems–a survey. *Data & Knowledge Engineering*, 50(1):9–34, 2004.

[100] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. Aalst. Process flexibility: A survey of contemporary approaches. *Advances in Enterprise Engineering I*, pages 16–30, 2008.

[101] Ragib Hasan, Radu Sion, and Marianne Winslett. Introducing secure provenance: problems and challenges. In *Proceedings of the 2007 ACM workshop on Storage security and survivability*, StorageSS '07, pages 13–18, New York, NY, USA, 2007. ACM.

[102] Umar Farooq and Wajeeha Khalil. Grid as Human? s Assistant: A Logical Solution Provider for Physical Problems. 2006.

[103] Emile Aarts and Reiner Wichert. Ambient intelligence. In Hans-Jörg Bullinger, editor, *Technology Guide*, pages 244–249. Springer Berlin Heidelberg, 2009. Accessible at: http://dx.doi.org/10.1007/978-3-540-88546-7_47.

[104] Accessible at: www.facebook.com.

[105] Accessible at: http://www.myspace.com/.

[106] Twitter. Accessible at: www.twitter.com.

[107] Blogger. Accessible at: https://www.blogger.com/start.

[108] Peter Paul Beran, Elizabeth Vinek, Erich Schikuta, and TThomas Weishaupl. Vinnsl - the vienna neural network specification language. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1872–1879, 2008.

[109] Erich Schikuta. Neuroweb: an internet-based neural network simulator. In *14th IEEE International Conference on Tools with Artificial Intelligence*, pages 407–412. IEEE, November 2002.

[110] Christian Brunner and Christian Schultes. Neuroaccess: The neural network data base system. Master's thesis, University of Vienna,Vienna, Austria, 1998.

[111] Juergen Mangler and Michael Derntl. CEWebS-Cooperative environment web services. In *Proceedings of 4th International Conference on Knowledge Management (I-KNOW04), June*, pages 617–624, 2004.

[112] Stefan Wurm. Reasoner based query machine. master thesis, Faculty of Computer Science, University of Vienna, December 2007.

[113] IEEE Computational Intelligent Society. Accessible at: http://www.ieee-cis.org/.

[114] Amit Konar. *Computational intelligence: principles, techniques and applications*. Springer Verlag, 2005.

[115] Janusz Wojtusiak, Jordan M. Malof Jacek M. Zurada, Devendra Mehta, and Khalid Moidu. Toward VO-based Collaboration between Computational Intelligence-Machine Learning and Healthcare Communities. *Intelligent Information Systems*, (9999):1–12.

[116] T. Weishaupl, C. Witzany, and E. Schikuta. gset: trust management and secure accounting for business in the grid. In *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*, volume 1, page 8 pp., may 2006.

[117] N2grid system. Accessible at: http://big.pri.univie.ac.at/n2grid/.

[118] Joy Pauschke, TL Anderson, SN Goldstein, and P. Nelson. Construction status of the George E. Brown, Jr. network for earthquake engineering simulation. In *Proceedings of the Seventh US National Conference on Earthquake Engineering (7NCEE)*, volume 21, page 25. Boston MA: July, 2002.

[119] TWGrid:Academia Sinica grid Computing. http://www.twgrid.org/.

[120] J. Mangler and M. Derntl. Cewebs-cooperative environment web services. In *Proc. 4th International Conference on Knowledge Management (I-KNOW'04), Graz, Austria*, pages 617–624. Citeseer, 2004.

[121] C.S. Ong, J.Y. Lai, and Y.S. Wang. Factors affecting engineers' acceptance of asynchronous e-learning systems in high-tech companies. *Information & management*, 41(6):795–804, 2004.

[122] M.H. Harun. Integrating e-Learning into the workplace. *Internet and Higher Education*, 4:301–310, 2001.

[123] C. Keller and L. Cernerud. Students perceptions of e-learning in university education. *Learning, Media and Technology*, 27(1):55–67, 2002.

[124] N. Wagner, K. Hassanein, and M. Head. Who is responsible for e-learning success in higher education? a stakeholders' analysis. *Analysis*, 11(3):26–36.

[125] D.Ö.G. Hoppe and M.H. Breitner. Business models for e-learning. 2003. Accessible at: http://cosmic.rrz.uni-hamburg.de/webcat/hwwa/edok04/de4826g/DP287.pdf.

[126] J.C. Cronje. Who killed e-learning? In *Proceedings of the Academic Libraries: Proactive Partners in Learning and Research Symposium at University of Stellenbosch, South Africa*, 2006.

[127] A.J. Romiszowski. How's the e-learning baby? factors leading to success or failure of an educational technology innovation. *EDUCATIONAL TECHNOLOGY-SADDLE BROOK THEN ENGLEWOOD CLIFFS NJ-*, 44(1):5–27, 2004.

[128] A. Wahlstedt. Stakeholders' conceptions of learning in learning management systems development. 2007.

[129] President's Information Technology Advisory Committee. Computational science: Ensuring america's competitiveness. Technical report, National Coordination Office for Information Technology Research and Development, Arlington, VA, June 2005.

[130] Eldad Eilam. *Reversing: Secrets of Reverse Engineering*. John Wiley & Sons, February 2011.

[131] Luis Camarinha-Matos, Hamideh Afsarmanesh, and Martin Ollus. *Virtual organizations: systems and practices*. Springer, 2005.

[132] M. Bakery and R. Buyyaz. Cluster computing at a glance. *High Performance Cluster Computing: Architectures and Systems*, pages 3–47, 1999.

[133] Luis Camarinha-Matos, Hamideh Afsarmanesh, and Martin Ollus. *Methods and Tools for Collaborative Networked Organizations*. Springer, 2008.

[134] Russell C. Eberhart and Yuhui Shi. *Computational intelligence: concepts to implementations*. Morgan Kaufmann, August 2007.

[135] E. Ferneley, M. Wetherill, and Y. Rezgui. Toward the construction knowledge economy: the e-cognos project. *Proceedings of ECIS 2002*, 2002.

[136] Xiaofei Xu and Zhongjie Wang. State of the art: business service and its impacts on manufacturing. *Journal of Intelligent Manufacturing*, 22:653–662, 2011. 10.1007/s10845-009-0325-3.

[137] C. Windham. nanohub: Educase learning institution (eli). 2007. Accessible at: http://net.educause.edu/ir/library/pdf/ELI3015.pdf.

[138] L. Bass, P. Clements, and R. Kazman. *Software architecture in practice*. Addison-Wesley Longman Publishing Co., Inc., 2003.

[139] Giuseppe Riva and Fabrizio Davide. *Communications through virtual technologies: identity, community and technology in the communication age*. IOS Press, January 2001.

[140] James Edward Smith and Ravi Nair. *Virtual machines: versatile platforms for systems and processes*. Elsevier, June 2005.

[141] P. Sieber and J. Griese. *Organizational virtualness*. Citeseer, 1998. Accessible at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.1293 &rep=rep1&type=pdf.