Technical University of Denmark

A Networked Perspective on the Engineering Design Process: At the Intersection of Process and Organisation Architectures

Parraguez Ruiz, Pedro; Maier, Anja

Link to article, DOI:
10.5281/zenodo.22785

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

The design process of engineering systems frequently involves hundreds of activities and people over long periods of time and is implemented through complex networks of information exchanges. Such socio-technical complexity makes design processes hard to manage, and as a result, engineering design projects often fail to be on time, on budget, and are not meeting specifications. Despite the wealth of process models available, previous approaches have been insufficient to provide a networked perspective that allows the challenging combination of organisational and process complexity to unfold. This thesis reports on research undertaken to develop, apply and test a framework that characterises the actual design process architecture of engineering systems as a networked process.

Pedro Parraguez
ppru@dtu.dk

DTU Management Engineering
Produktionstorvet Building 424
2800, Kongens Lyngby
Tlf. +45 4525 4800

http://www.man.dtu.dk/

ISBN 978-87-998410-0-4

Pedro Parraguez

A Networked Perspective on the Engineering Design Process

A Networked Perspective on the Engineering Design Process:

At the Intersection of Process and Organisation Architectures

Pedro Parraguez
PhD Thesis, March 2015
Engineering Systems Group

DTU Management Engineering
Department of Management Engineering

Technical University of Denmark

A Networked Perspective on the Engineering Design Process:
At the Intersection of Process and Organisation Architectures

Pedro Parraguez

PhD Thesis, March 2015

Engineering Systems Group
DTU Management Engineering

ISBN: 978-87-998410-0-4

To my family in Chile and all those who with their continuous support and inspiration made this journey possible

Abstract

The design process of engineering systems frequently involves hundreds of activities and people over long periods of time and is implemented through complex networks of information exchanges. Such socio-technical complexity makes design processes hard to manage, and as a result, engineering design projects often fail to be on time, on budget, and meeting specifications. Despite the wealth of process models available, previous approaches have been insufficient to provide a networked perspective that allows the challenging combination of organisational and process complexity to unfold. The lack of a networked perspective also has limited the study of the relationships between process complexity and process performance. This thesis argues that to understand and improve design processes, we must look beyond the planned process and unfold the network structure and composition that actually implement the process. This combination of process structure—how people and activities are connected—and composition—the functional diversity of the groups participating in the process—is referred to as the actual design process architecture.

This thesis reports on research undertaken to develop, apply and test a framework that characterises the actual design process architecture of engineering systems as a networked process. Research described in this thesis involved literature reviews in Engineering Design, Engineering Systems, Complexity and applied Network Science, and two case studies at engineering design companies with the objective of iteratively developing the framework and providing a proof-of-concept of its use in a large engineering design project.

The developed Networked Process (NPr) Framework is composed of a conceptual model of the actual design process architecture, and an analytical method that allows the model and data-driven support to be quantified. The framework provides a networked perspective on three fundamental levels of analysis: 1) the activity-level, characterised as a network of people performing each activity, 2) the interface-level, characterised as a network of people interfacing between two interdependent activities, and 3) the whole process-level, characterised as a dynamic network of people and activities. The aim of the framework is to improve the design process of engineering systems through a more detailed overview of the actual design process, to support data-driven reflection of the relationship between process architecture and performance, and to provide the means to compare process plans against the actual process. The framework is based on a multi-domain network approach to process architecture and draws on previous research using matrix-based and graph-based process models.

The results of the NPr Framework's application in two case studies showed that decision makers in engineering design projects were able to gain new insights into their complex design processes through the framework. Such insights allowed them to better support and manage design activities, process interfaces and the whole design process. The framework also was used to enrich project debriefing and lessons-learned sessions, to spot process anomalies, to improve design process planning, to examine process progress, and to identify relationships between process architecture and performance. Contributions to knowledge include: First, the development of a more complete model of the actual process architecture and concrete analytical methods to quantify the developed model. Second, the identification of key structural and compositional variables as well as tests to identify the relationship between those variables and performance metrics. Third, the creation of a platform for further research on the relationships between actual design process architecture, behaviour and performance.

PhD supervisor

Professor Anja M. Maier
Engineering Systems Group
DTU Management Engineering
Technical University of Denmark

Examiners

Professor Tim C. McAloone
Engineering Design and Product Development
DTU Mechanical Engineering
Technical University of Denmark

Associate Professor Manuel Sosa
Technology and Operations Management
INSEAD Singapore

Professor Martin Steinert
Engineering Design and Innovation
Department of Engineering Design and Materials
Norwegian University of Science and Technology

Acknowledgements

I'm grateful to many people for their invaluable contributions and support throughout my research journey. This thesis greatly benefited from their experience and kind feedback. I feel truly privileged for having worked alongside them. First and foremost, I would like to thank my supervisor, Anja Maier, for continuously going above and beyond her responsibilities. Her trust, guidance, encouragement, and patience allowed me to grow both academically and personally. I feel very fortunate for having had her as my PhD supervisor.

I would also like to thank all my colleagues and friends in the Engineering Systems Group at DTU Management Engineering, as well as many others in DTU at large. It has been a pleasure to share this special time of my life with them. Their professional support and friendship made me feel at home, and kept me sane during the many ups and downs, which are so common of academic research.

Very special thanks go to Steven Eppinger, who opened the doors of MIT and contributed to this research at key stages with his knowledgeable insights and advice. Working with him has been an honour and being a visiting researcher at MIT, the birthplace of so much Engineering Systems research and leading scholars, a truly unique opportunity.

During this PhD project I benefited from the advice and feedback of many who kindly heard and read earlier versions of this work. I'm particularly grateful to Alison Olechowski, Christian Sørup, Erika Buonansegna, Johnny Marcher, Maria Tennant, Melanie Kreye, Mogens Myrup Andreasen, Nathan Crilly, Nicholas Kristensen, Peter Dolf, Phil Cash, and Thomas Howard for their honest inputs, ideas and corrections. I would also like to recognise the influence of Chris McMahon as my MSc dissertation advisor at the University of Bath. He introduced me to the fascinating Engineering Design community and supported my first steps in the world of academic research, kick-starting this new career path.

Last but not least, this research would not have been possible without the generous support and contributions in time and data of the two companies that participated as case studies. Their participation and trust provided essential empirical grounding for this thesis. Furthermore, I would also like to express my appreciation to the Otto Mønsteds Fond and Idella Foundation for the research grants that funded academic conferences and my research abroad at MIT.

—Thank you all.

The following publications are part of the work presented in this thesis:

Conference papers and abstracts:

Parraguez, P. & Maier, A.M., 2015. Unfolding the Design Process Architecture: A Networked Perspective on Activities. In *International Conference on Engineering Design 2015 (ICED 2015)*. Milan.

Parraguez, P., Eppinger, S.D. & Maier, A.M., 2014. Evolution of information control and centralisation through stages of complex engineering design projects. In M. Dorian et al., eds. *Proceedings of the 13th International Design Conference (DESIGN 2014)*. Dubrovnik.

Parraguez, P. & Maier, A.M., 2013b. Information Flows in Networked Engineering Design Projects. In *5th Workshop and Conference on Information in Networks (WIN)*. New York.

Parraguez, P. & Maier, A.M., 2013a. An Activity-Centric Network Model of Engineering Design as a Social Process. In *NetSci 13'*. Copenhagen.

Journal papers under review:

Parraguez, P., Eppinger, S.D. & Maier, A.M., 2014b. *Information Flow through Stages of Complex Engineering Design Projects: A Dynamic Network Analysis Approach.* Revise and re-submit to IEEE Transactions on Engineering Management.

Parraguez, P., Eppinger, S.D. & Maier, A.M., 2014a. *Characterising design process interfaces as organisation networks: Insights for engineering systems management.* Revise and re-submit to Systems Engineering

Other publications produced during the PhD project but not presented in this thesis:

Parraguez, P. & Maier, A.M., 2012b. Towards describing co-design by the integration of Engineering Design and Technology and Innovation Management literature. In P. K. Hansen et al., eds. *NordDesign Conference 2012*. Aalborg: Center for Industrial Production, Aalborg University, pp. 465–481.

Parraguez, P. & Maier, A.M., 2012a. Mapping industrial networks as an approach to identify inter-organisational collaborative potential in new product development. In C. Hernández-Cuevas et al., eds. *Encuentros Paris 2012 – Knowledge for Economic and Social Development*. Paris: Encuentros.

Table of contents

1 INTRODUCTION

Perplexity is the beginning of knowledge
—Khalil Gibran

This thesis presents a novel framework composed of a conceptual model, a set of analytical methods, and data-driven support for the design process of engineering systems. The framework allows the network architecture of the actual design process to be quantitatively characterised, the relationships between the actual design process architecture and process performance to be tested, and the planned design process to be compared with the actual design process architecture. The aim of this framework is to provide a better understanding of the actual design process architecture and to support the design process of engineering systems. A premise is that the network architecture of the actual design process has real-world properties and has influence on the results of the design process.

This chapter gives an introduction to the PhD thesis, starting with the overall motivation and problem definition (section 1.1), which highlight the challenges of designing engineering systems and the current industrial needs and knowledge gaps. Section 1.2 lays out the research objectives and the main research questions, based on the need and knowledge gaps introduced in the previous section. Section 1.3 describes the research scope and underlying premises of this research. Section 1.4 gives a brief overview of the overall research approach, and finally, section 1.5 concludes with an overview of the thesis structure.

1.1 Motivation and problem definition

Engineering projects often fail to be on time, on budget, and on specifications, particularly when their goal is the design of engineering systems (de Weck et al., 2011, p. 34). In fact, the Project Management Institute (PMI) estimated that 44% of engineering system projects fail to meet their goals. *'This poor performance results in organizations losing $109 million for every $1 billion invested in projects'* (PMI, 2014). Besides the substantial economic impact of projects running over timelines and budgets, engineering systems that fail to meet design specifications potentially could affect the lives of thousands or millions of people, and could generate serious and unexpected externalities.

Iconic examples of design failures in engineering systems include the battery problems in the Boeing 787, the blowout preventer failure in Deepwater Horizon, and the critical failure of O-rings in the Challenger space shuttle. All of these problems emerged in engineering design processes staffed with some of the best minds available, using state-of-the-art technology, and applying some of the most advanced design processes and practices. However, despite all their resources, these projects still suffered from critical, yet in hindsight avoidable design process failures.

For instance, in the case of the Boeing 787, the final report from the Critical Systems Review Team (Federal Aviation Administration, 2014) found that the primary cause of system failures was not the novelty of the technologies, but the design process by which these technologies were integrated. Among the highlighted design process problems, the review noted 'inadequate communication of the requirements', 'unclear ownership of design requirements', 'established design review process not being followed when design requirements cross organizational or design boundaries', and 'inadequate design requirements due to incorrect assumptions about how the designed systems would perform'. In the case of the Deepwater Horizon, the safety board investigator concluded that a key problem was that 'well owner BP and rig operator Transocean didn't test the blowout preventer's individual safety systems. They just tested the device as a whole' (The Guardian, 2014), showing a lack of coordination between stakeholders in a key process activity. In turn, the Rogers Commission concluded that the cause of the Challenger Space Shuttle disaster 'was due to a faulty design unacceptably sensitive to a number of factors' combined with serious communication problems concerning 'incomplete and sometimes misleading information', and 'a conflict between engineering data and management judgments' (Rogers Commission, 1986). These examples point to the importance of examining the design process of engineering systems, in particular, the way in which people shape and implement complex design processes.

Although the three previous engineering systems are of exceptionally large scale, the scale of engineering systems can range widely. For instance, engineering systems can go from entire national energy and transportation systems, to power plants, complex processes such as the Toyota Production System (TPS), and next-generation water filtration membranes. As the subject of this thesis is the design process of engineering systems, I use the degree of socio-technical complexity in a design process to draw the boundary between what it is an engineering system and what is not. This socio-technical complexity is a result of the many social and technical elements, multiple levels of system decomposition, numerous interactions, and various interdependencies between the elements found in the engineering system (de Weck et al., 2011, p. 31). Such combination of features makes engineering systems inherently more ‘difficult to describe, understand, predict, manage, design, and/or change’ (Magee and de Weck, 2004). For example, a component in a water filtration plant could be examined in isolation and its product architecture described by only a few elements and their interactions; however, the same component can become an engineering system if the components with which it interacts and the machinery required to build such components are analysed as a system. This complexity also makes an accurate process overview unfeasible for individuals who lack additional support to structure the rich stream of information produced during the process. These difficulties and complexities make the design process of engineering systems a challenging endeavour and help to explain the high proportion of engineering system projects that fail to be on time, on budget, and on specifications.

1.1.1 The design process of engineering systems

The design process of engineering systems can easily involve hundreds or thousands of geographically distributed project members, require deep and diverse technological expertise, span several years, and involve multiple organisations interacting in large development networks. All these features set these processes apart in terms of complexity and increase the challenges and needs related to process overview, systems integration, and communication (Madni and Sievers, 2014). Based on this understanding, extensive research efforts have been devoted to increase our knowledge about complex engineering systems as well as the design process of these systems. For example, research in the fields of Engineering Systems and Engineering Design has found that the way in which engineering systems are structured and composed, that is, their product architecture, is an important determinant of their performance and vulnerabilities (Crawley et al., 2004; Sosa et al., 2011).

As with products and organisations, the performance of the design process of engineering systems can be examined in terms of efficiency and effectiveness. The efficiency of the design process relates to the ability to deliver the required design outputs on time and on budget; its effectiveness relates to the quality of the final design output, or in other words, the ability of the process to deliver the desired specifications and the innovativeness of the results (O’Donnell and Duffy, 2002). While there are means to test for relationships between aspects such as product architecture and product performance (e.g. Yassine & Wissmann 2007; Sosa et al. 2007a), no equivalent advances have been made to methodically characterise the actual process architecture and test for relationships between that process architecture and its performance (Kreimeyer and Lindemann, 2011, p. 20).

The actual process architecture is difficult to characterise because, unlike the planned process architecture, the actual architecture cannot be modelled based on data about technical information dependencies between tasks or process plans, but requires data about the process that actually happens. For example, this data includes information about the network of information exchanges among people performing design activities throughout the development process. Although such data often exists as digital traces, it is harder to structure because it involves working at the intersection of process and organisation architectures, which requires new models and methods.

This thesis will show that the ability to characterise the networked design process is needed to fill current knowledge gaps and better support the design process of engineering systems. The rationale is that through a systematic characterisation of the actual design process architecture, it becomes possible to gain enhanced process overview, identify relationships between the design process architecture and process performance, obtain valuable feedback for design process improvements though an active comparison of actual and planned processes and ultimately, design better processes and support mechanisms to improve the design process of engineering systems.

1.1.2 The design process architecture

During the design process of engineering systems, three interconnected domains are found, each with its own architecture: the product, the organisation and the process domains (Browning, 2001; Eppinger and Browning, 2012). *The product domain* refers to the engineering system that is being designed, such as a biomass power plant, with an architecture defined by its technical components (and subsystems) and the interconnection of the components. *The organisation domain* refers to the organisation or group of organisations and other stakeholders involved in the design process, with an architecture defined by people and their interactions. *The process domain*, the central focus of this thesis, refers to the series of activities or tasks needed to design the engineering system. The process domain has an architecture often characterised as an activity or a task network, connected by output-to-input information relationships between its elements (Browning and Ramasesh, 2007). Studies of the design process usually focus on one or more of these domains, and some explicitly consider cross-domain interactions between the architectures of these domains (e.g. Eppinger & Salminen 2001; Maurer 2007; Bartolomei et al. 2012).

Two conceptual building blocks used in this thesis to characterise the actual process architecture are the **network structure and composition** of the process architecture. In turn, **three fundamental levels of analysis** are used to study the process architecture, the activity, the interface, and the whole activity network levels.

- **Network structure and composition**: In network models, the constituent elements (activities) and the way in which these elements are connected (information flows) are referred to as the **architecture** of the network[1] (IEEE Standards Board, 2000). This architecture can be examined through the network **structure**, the arrangement of and relationships among the elements of the network, and the network **composition**, the types and/or features of the elements that constitute the network (Wasserman and Faust, 1994).
- **Three levels of analysis:** Any network architecture can be analysed at three basic levels of analysis: 1) the level where each element is represented as a unique node, 2) the level where the connection between two elements is represented as an edge, and 3) the level that contains the combination of elements and their connections, represented as a the whole network (Borgatti and Foster, 2003; Wasserman and Faust, 1994). In the context of activity-based network models of process architecture, the equivalent levels are: **activities**, **interfaces** that allow for the information flows between activities, and the **whole activity network**. Although these levels of analysis are intrinsically interdependent, they can also be analysed independently in relationship to performance metrics obtained at their respective levels.

As illustrated in Figure 1-1, these building blocks are not exclusive to process architecture, but also can be applied to the architectures in the product and the organisation domains to yield a more complete and connected architectural perspective. Such a networked perspective allows for differentiation between a process architecture modelled as a network of tasks connected by technical information dependencies and one modelled as a network of activities connected by information flows. The information dependencies between tasks result from parameter interdependencies between the engineering system components and create the intersection between the product and the process architectures. In turn, the information flows

[1] Here network and system architecture are used synonymously. The main distinction is that a network architecture is one possible model of a system architecture, which is in turn a representation of a real-world phenomenon (Baldwin et al., 2013).

between activities emerge from information exchanges between people and create the intersection between the process and the organisation architectures, which is the focus of this thesis.

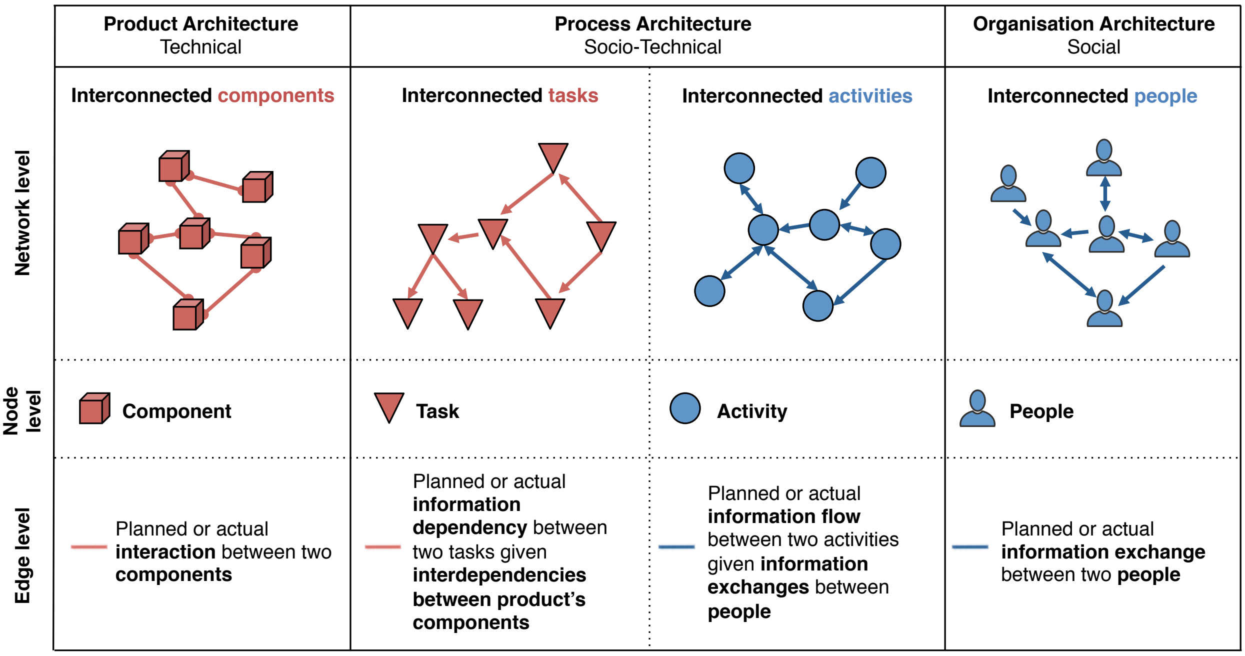


Figure 1-1: Summary description for the product, the process, and the organisation

1.1.3 Identified needs and knowledge gaps

The motivation for this research stems from unresolved challenges hindering the performance of the design process of engineering systems and from limitations in the models and methods available to study the actual process architecture. The most important needs and knowledge gaps identified through the literature review and the performed exploratory fieldwork can be divided into three guiding topics:

1) Conceptual characterisation of the actual design process architecture

- **Industrial need:** Lack of overview about the actual design process and fragmentation of current process models.

  A number of process models and views are available to describe and plan the design process inside organisations (Browning, 2009), including Gantt charts, Design Structure Matrices (DSMs), PERT diagrams, and flowcharts. Yet, these are infrequently updated and provide a fragmented view. Moreover, these methods represent what the company plans, believes, or expects to happen, and do not provide a window into the actual design process. This lack of systemic overview of the actual design process leaves organisations designing engineering systems without the required support to visualise their on-going activities and to learn from their own design patterns. Previous studies on design communication have highlighted the importance of factors such as understanding information needs, an overview of task sequence and task handover, and clarity of roles and responsibilities (Maier et al., 2008; Maier, Kreimeyer, et al., 2009). However, in large engineering design projects, obtaining current, bottom-up overviews of the actual process—which could enhance the maturity level of these factors—remains an open challenge.

- **Knowledge gap:** Current models and methods are insufficient to characterise the actual design process architecture.

  Although a wide variety of design process models exist, most are focused on providing prescriptions for suggested stages and activities, represent the planned process or are based on static, top-down estimations of the process (Clarkson and Eckert, 2005, p. 21). Therefore, process models can work as useful guidelines or benchmarks, but have limitations when it comes to capturing actual designing patterns (Clarkson and Eckert, 2005, p. 18; Simon, 1946). The literature review in chapter 2 identifies models and methods for characterising aspects of the actual design process (e.g. Durugbo et al. 2011; Maurer 2007; Morelli et al. 1995; Sosa 2014; Clarkson & Hamilton 2000), but does not include a satisfactory approach to activity networks that simultaneously considers multiple levels of analysis, network structure and composition, and process dynamics, and that can be used to study large-scale design processes in real time.

2) Quantitative characterisation of the design process architecture that enables the comparison between the planned and the actual process architecture

- **Industrial need:** A comparison of the planned design process with the actual design process and its progress is difficult to make.

  Without means to follow and describe their actual design processes, organisations cannot benchmark or compare their plans and expectations against their real engineering design work. This problem hinders a key feedback mechanism that could allow project managers and design engineers to exercise 'reflection-in-action' (Schön, 1984), and is especially restricting when it comes to observing and reflecting on the consequences of activities and interactions that happen outside each project participant's local and limited awareness.

- **Knowledge gap:** Conceptual constraints for the comparison between planned and actual engineering design processes.

  To date, a quantitative and systematic comparison between the planned and the actual architectures of the design process is not possible because of insufficient means to characterise the actual process. As a result, testing design process theories and models is limited because comparisons between a theoretical model and an actual process require being able to quantify the actual process and to translate it into a comparable representation.

  Two types of studies demonstrate the usefulness of analysing and comparing domain architectures. The first is composed of research testing the 'mirroring hypothesis', analysing the degree of alignment between the architectures of the process, product, and/or organisation domains (e.g. Sosa et al. 2004; MacCormack et al. 2012; Colfer & Baldwin 2010). The second type analyses differences between the formal and informal architecture of the organisation domain (e.g. Allen et al. 2007; Kratzer et al. 2008; Labianca 2004). What is missing is an approach that would allow a comparative analysis between the planned and the actual process architecture.

3) Data-driven evidence and support

- **Industrial need**: Generic prescriptive advice has a limited use when industry, designed systems, and organisational characteristics differ so widely.

  Although generic advice based on in-depth studies of design processes across a range of organisations and industries are available–and there is evidence they can improve engineering design practice (Roozenburg and Cross, 1991)–project managers and design engineers also must reflect on and assess their own practices (Maier, Kreimeyer, et al., 2009; Schön, 1984). Furthermore, what works in one organisation might not work elsewhere, and even could be counterproductive. Therefore, to complement already available general guidelines, new approaches are needed that use real design process data and allow connections to be made between the company's execution of the design process and metrics of efficiency and effectiveness.

- **Knowledge gap:** Need for sufficient variability to establish relationships between design process architecture and performance.

  To identify consistent relationships between design process architecture and process performance, the dependent and independent variables under analysis must have sufficient variability. At the same time, other exogenous variables must exhibit a minimum of variation, or be subject to controls (March and Sutton, 1997). Unfortunately, the highly contingent nature of the design process limits comparisons across various organisations (Bucciarelli, 1988). Therefore, new approaches are required that allow establishing reliable links between actual process architecture and performance for each design process context.

1.2 Research aim, research objectives, and main research questions

Research Aim:

The overall aim of this thesis is to provide (i) the **means to characterise the actual design process architecture** and (ii) **data-driven support to the design process of engineering systems**.

As such, the research aim is divided into two parts:

i. A **descriptive** part with the goal of improving current understanding of the actual design process architecture. This part is implemented through a conceptual model and a set of analytical methods to quantitatively characterise the actual network architecture of engineering design processes. This part fills identified knowledge gaps.

ii. A **prescriptive** part with the goal of using the understanding gained from part (i) to support the design process and to prescribe what organisations can do to benefit from this knowledge. This part meets the identified industrial needs.

The developed framework provides an integrated response to satisfy both research aims. Based on the previously introduced guiding topics (section 1.1.3) the overall research aim has the following three research objectives.

Research objectives:

- RO1: To develop a multilevel, dynamic **characterisation of the actual design process architecture**
- RO2: To **enable the comparison between the actual and the planned design process** architectures
- RO3: To provide **means for connecting the characterisation of the actual design process architecture with process performance metrics** to support design process improvements

These research objectives were operationalized into key research questions. For each research question, introduced below, section 3.2.3 presents concrete success criteria to evaluate the answers provided by this thesis to those questions (presented in section 3.2.3) were developed with which to against which the measure the answers provided to those questions could be measured.

Main research questions

This research was structured around three main research questions. The focus was first to develop conceptual (RQ1) and quantitative (RQ2) characterisations to describe the actual design process architecture, and second to develop prescriptive support through such characterisations (RQ3).

Research questions:

- RQ1: **How can we model** the multilevel, dynamic, and actual design process architecture of engineering systems?
- RQ2: **How can we quantitatively characterise** the model of the actual architecture so that it is analytically comparable to planned process architecture views of engineering systems?
- RQ3: **How can we connect a quantitative characterisation** of the actual architecture **with process performance metrics?**

Figure 1-2 connects and summarises the guiding needs and knowledge gaps, research objectives, research questions, and the expected outputs of answering the posed research questions.

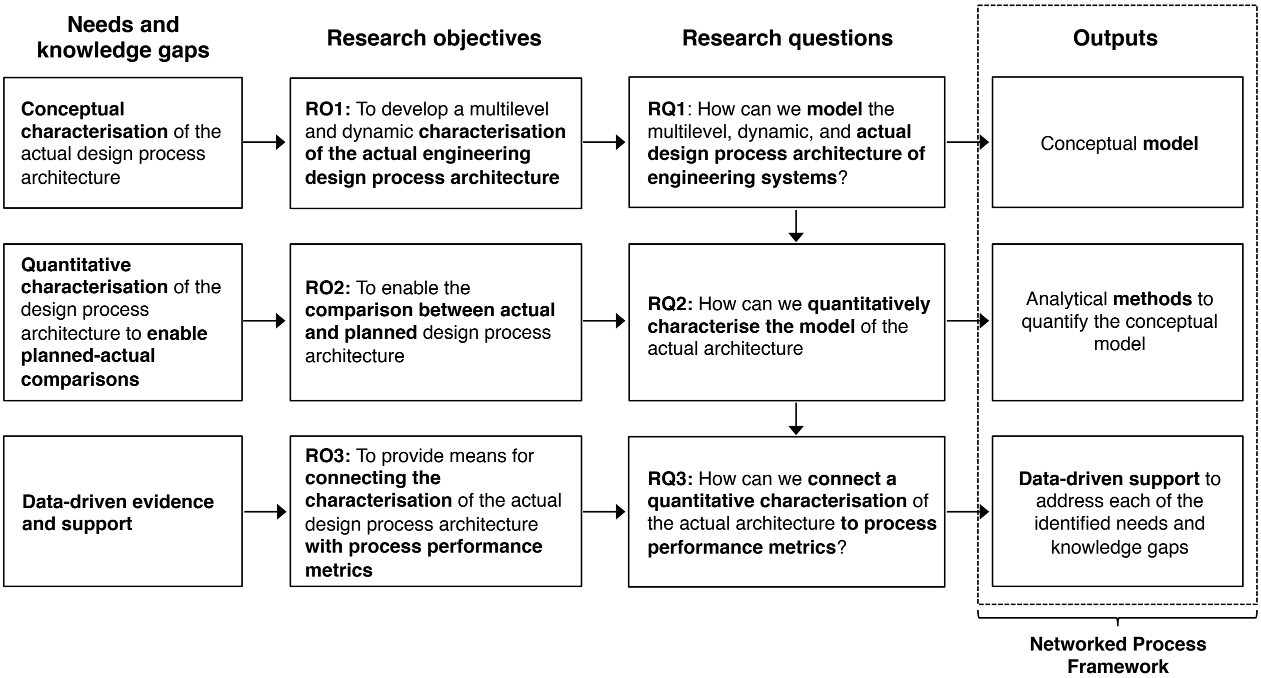


Figure 1-2: Relationships between needs and knowledge gaps, research objectives, research questions, and expected outputs.

1.3 Research scope and underlying premises

The focus of this thesis is on the actual design process of engineering systems when analysed at the intersection of process and organisation architectures. This intersection centres the attention on people, activities, and information flows. To achieve the defined research objectives, this thesis develops a networked perspective of the engineering design process, materialised in a framework that comprises a conceptual model of the actual process architecture and an analytical method that allows for a quantification of the conceptual model.

In terms of industrial applications, the emphasis is on the design process of engineering systems. In such systems, the scale and degree of socio-technical complexity challenges our theoretical understanding, conventional project management tools, and means to support the design process. In general, the higher the socio-technical complexity of the design process, the more relevant becomes the approach this thesis proposes.

Outside the scope of this thesis are cognitive level processes, an analysis of the product architecture, process simulations, and optimisation methods of the design process architecture, such as sequencing or partitioning. Although the proposed model of the actual process architecture and its characterisation could be conceptually examined through the theoretical lenses offered by Activity Theory, Actor Network Theory, Distributed Cognition, Situated Cognition, and related approaches, this work's scope and practical limitations do not allow for explicit consideration or discussion of those approaches.

The thesis is built on two premises. The first premise is that **the design process of engineering systems is a complex socio-technical system of information transformation**. The social complexity of the process results from the rich information-driven interactions between project members observed in information exchanges. These interactions are assumed to be essential to transform a set of requirements and pre-existent knowledge (information inputs) into detailed designs (information outputs). This view of design as a social process is consistent with that of many researchers in engineering design, with prominent examples found in the work of Minneman (1991), Bucciarelli (1988), Schön (1984), and Simon (1996). The process's technical complexity arises from the combination of many interdependent design tasks, the engineering system being designed, and the multiple enabling technologies utilised during the design. Such a view of design as a complex technical process is manifest in the extensive work on structural complexity management, especially in the product and process domain (Eppinger and Browning, 2012; Lindemann et al., 2009; Minai et al., 2006).

The second premise is that **the architecture of the design process, that is, its structure and composition, generates design process behaviour**. In turn, the expressed behaviour determines the process performance and the designed engineering system. This premise is consistent with the Function-Behaviour-Structure theory (Gero and Kannengiesser, 2002) and particularly its extension to general processes (Gero and Kannengiesser, 2007). More generally, this premise is also the foundation of Network Science, which maintains that a wide range of real-world phenomena are affected and sometimes produced by the network architecture of the systems where such phenomena emerge (Borgatti and Halgin, 2011; Strogatz, 2001).

1.4 Research approach

As an overall approach, this thesis uses the Design Research Methodology (DRM) (Blessing and Chakrabarti, 2009), following its stages of research clarification, descriptive study I, prescriptive study, and an initial descriptive study II. The research methods used to develop the framework combine quantitative and qualitative approaches for data acquisition, analysis, and interpretation. Network analysis methods were used extensively, and therefore, data acquisition was focused on relational information. Due to the nature of the developed framework, at the intersection of process and organisation architectures, most of the gathered relational data was about people's information exchanges and their participation in activities.

The elicitation of requirements, data gathering, network analysis, and testing of the conceptual model and analytical methods were performed through two industry case studies. The first was used as an exploratory case to develop the framework and to pilot its application. The second was used as a descriptive case to apply and evaluate the final version of the framework within a larger, more complex engineering design project. The quantitative results of this second case study and its interpretation are included in full in this thesis.

1.5 Thesis structure

The reminder of this thesis is structured as follows:

- **Chapter 2** provides a literature review focused on network-based approaches to the design process of engineering systems. The review also examines essential background about complexity and network science to allow the characterisation of process architectures. This chapter concludes with the identification of literature gaps, further elaborating on the knowledge gaps identified in this introduction.
- **Chapter 3** describes the employed research methodology, including a more detailed breakdown of the research questions, the research approach to develop the framework, a description of the research methodology stages, and information about the two empirical studies.
- **Chapter 4** develops a multilevel framework that provides a networked perspective on the engineering design process, encompassing the architectures of the actual design process at the levels of activities, interfaces, and the whole process. The framework is divided into a conceptual model and an analytical method to allow the quantification of the proposed model.
- **Chapter 5** applies the developed framework to each level of analysis, utilising empirical data from the descriptive case study. This chapter provides a concrete proof-of-concept for the framework, demonstrating its practical application using real-world data and showing expected insights. Each main section of this chapter closes with a discussion of the obtained results from the perspective of the case study.

- **Chapter 6** evaluates and discusses the framework based on the gaps identified in sections 1.1.3 and 2.6, as well as the research objectives and success criteria. This chapter evaluates and discusses the developed framework's ability to answer the research questions and address the identified industrial needs and knowledge gaps.
- **Chapter 7** concludes and summarises this thesis and includes a reflection on theoretical and industrial contributions, managerial implications, limitations, and future work.

Finally, this thesis provides **a set of appendices** including a glossary of key terms and supplementary material. Figure 1.3 provides a graphical guide for the contents of this thesis, highlighting the key topics and illustrating the relationships between the seven chapters.

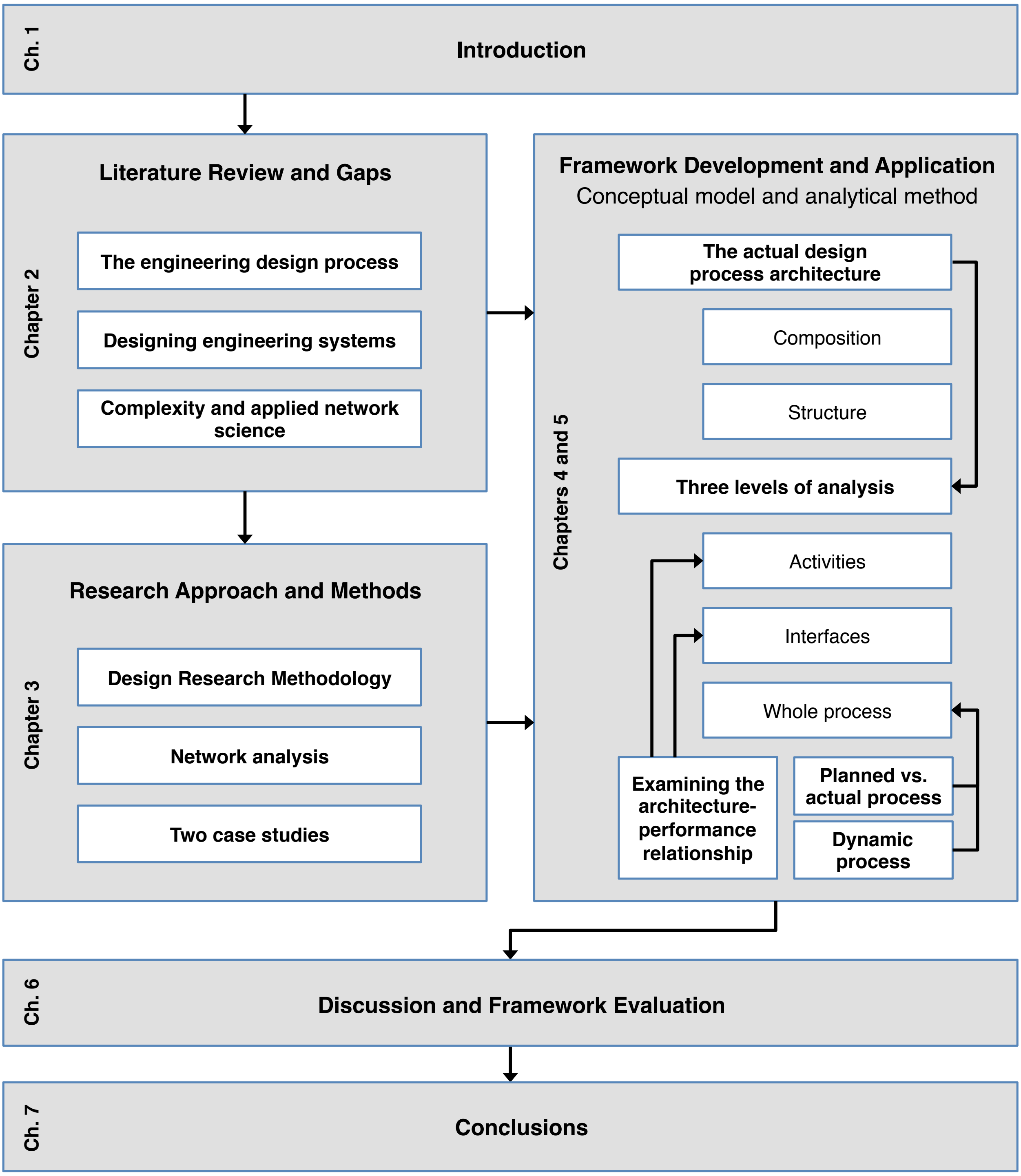


Figure 1-3: Thesis structure, key content, and relationships between chapters

2 LITERATURE REVIEW

All teaching and all intellectual learning come about from already existing knowledge

—Aristotle

The design process of engineering systems can be studied from diverse disciplinary angles. For example, contributions and valuable insights have been developed in research fields such as Organisational Science (Dougherty, 2008; Raisch et al., 2009), Information Science (Durugbo et al., 2011; Sundararajan et al., 2013), Operations Research (Evans and Jukes, 2000; Malone and Smith, 1988), Technology and Innovation Management (Gupta et al., 1985; Moenaert et al., 2000), and Knowledge Management (Behrend and Erwee, 2009; Jerome, 2012). However, based on the motivations, objectives, and scope of this research, this literature review focuses primarily on three research areas: 1) Engineering Design, in particular studies of the design process, 2) Complex Socio-Technical Systems, in particular studies on engineering systems, and 3) Complexity and Network Science, in particular network-based approaches to characterise the architecture of complex systems. This literature is synthesised as **the study of the design process of engineering systems utilising network-based approaches**, and illustrated in figure 2-1.

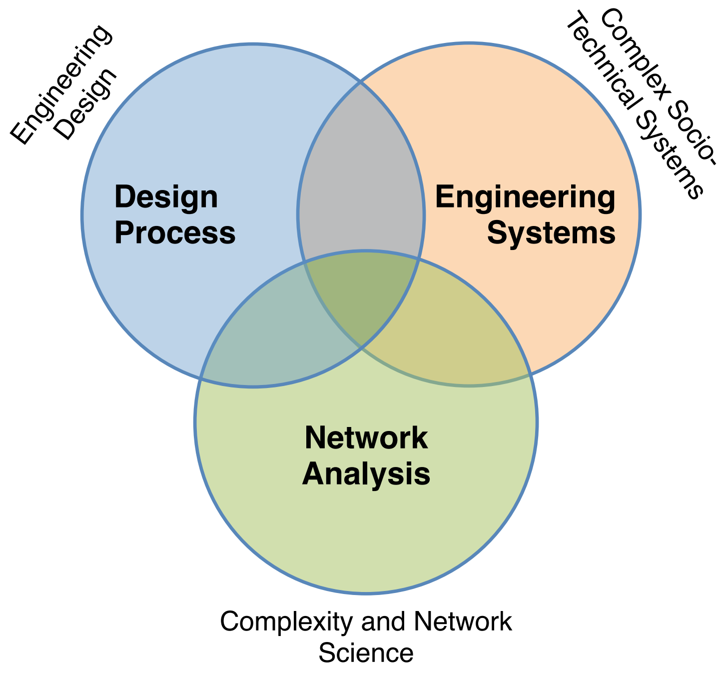


Figure 2-1: Core research areas

The field of Engineering Design constitutes the main literature source for the design process, and is where this thesis aims its contribution. The strengths of Engineering Design literature lie in its deep understanding of the design process. However, the challenges associated with the increased socio-technical complexity of designing engineering systems and analysing networks fall outside the core expertise of Engineering Design. To fill that gap, this thesis draws on knowledge from the fields of Engineering Systems, Complexity, and Network Science. Combining knowledge about the design process with knowledge about engineering systems can produce a detailed understanding of the distinctive nature and challenges of designing engineering systems. If we subsequently combine knowledge about the design process of engineering systems with knowledge from studies of complexity and networks, we can obtain a number of network properties and methods to quantitatively characterise design as a complex socio-technical system.

In addition to the core disciplines mentioned, studies that have connected organisation architecture with performance indicators of efficiency and effectiveness also provide evidence of architecture-performance relationships and were used to interpret this research's empirical findings (see section 2.2.5). Such studies are dispersed in the larger body of management-related research.

This chapter combines the key topics and academic fields previously enunciated. Section 2.1 describes the engineering design process, bringing together Engineering Design and Engineering Systems. Section 2.2 explores complexity and applied network science contextualised in the design of engineering systems. Section 2.3 reviews and structures theory of the process domain, and section 2.4 does the same for the organisational and product domains. Section 2.5 reviews existing literature to explore the intersection between the process, organisational, and product domains. Finally, section 2.6 identifies literature gaps that must be addressed to answer the research questions appropriately.

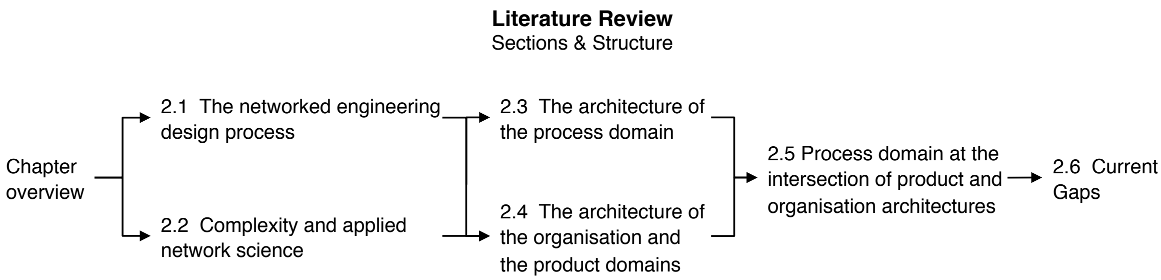


Figure 2-2: Structure of the literature review

2.1 The networked engineering design process

This thesis follows the view that engineering **design is a social process of information transformation** by which information-driven interactions among design engineers and other participants transform a set of requirements into detailed specifications. In addition, for engineering systems, the design process is modelled and analysed as a **complex socio-technical system of information transformation**, explicitly integrating the dimensions of social and technical complexity. The emphasis here is on the *actual* design process at the intersection of process and organisation architectures, and more specifically, the description of **design process patterns from a network perspective**.

Studying design as a socio-technical process of information transformation embodies two not always explicitly connected but intrinsically related views found in the Engineering Design literature: the view of design as a social process of information transformation and the view of design as a technical process centred on a sequence of interdependent tasks with a problem-solving focus. By integrating these two views, we will see how the networked perspective of the engineering design process emerges, and how the social and technical aspects can be analysed as an integrated whole.

2.1.1 Design as a social process

…(Design) exists only in a collective sense. Its state is not in the possession of any one individual to describe or completely define, although participants have their own individual views, their own images and thoughts, their own sketches, lists, diagrams, analyses, precedents, pieces of hardware, and now spread-sheets which they construe as the design. This is the strong sense of design is a social process.

(Bucciarelli, 1988)

Design has been considered a social process (Bucciarelli, 1988; Cross and Cross, 1995; Kleinsmann et al., 2007; Maier et al., 2005; Maier, Kreimeyer, et al., 2009; Minneman, 1991; Schön, 1984) with communication at the heart of design's coordination (Maier et al., 2008). Therefore, understanding this social dimension of design and design communication is essential for design process improvements (Eckert et al., 2005; Maier et al., 2005).

The various theories that have emerged to analyse design communication can be categorised based on their respective focus on information, interaction, situation, or a systemic view integrating all of those aspects (Eckert et al., 2005; Maier et al., 2005). In the systemic view, communication is described as a rich, interactive, and dynamic social process. Despite this comprehensive picture of communication, operationalising the systemic view to support industrial design processes is challenging. Because of the difficulty of directly applying systemic communication theories to organisational or project diagnoses, alternative approaches have been developed to operationalise empirical research about communication. These strategies usually involve utilisation of proxies, such as information exchanges or flows (e.g. Yassine et al. 2008), social interactions (e.g. Felekoglu et al. 2013), and/or the assessment of communication through systematic identification of key influencing factors (e.g. Maier et al. 2006). A common thread throughout research in this area is the idea of information, information exchanges, and information flows.

In engineering design, **information** is often the result of a combination of design inputs and outputs in the form of written documents, conversations, visual representations, gestures, and so on (Maier, Kreimeyer, et al., 2009). In the context of a design activity, this information is used to define a parameter, evaluate design options, and/or manage the design process (Sim and Duffy, 2003). Tribelsky and Sacks (2010) named a single piece of information about a design parameter (dimensions, weight, amount) as an ‘information item’, and defined ‘information package’ as the set of related information items that can take the form of a drawing, a worksheet, a document, a presentation, and so on (Tribelsky and Sacks, 2010). Given the social nature of the design process, *information exchanges* and *information flows* have been frequently used to characterise the design process, terms that simultaneously address the informational and interactional aspects of communication theories (e.g. Steward 1981; Yassine, Chelst, et al. 1999).

Elaborating on these ideas and following Tribelsky and Sacks (2010), an **information exchange** is defined as a communication event in which an information package containing a set of information items is transmitted between parties of the design process at a particular point of time. In turn, **an information flow** is a combination of information exchanges, or more precisely, a set of information packages exchanged between designers within or between design activities over a defined period of time.

With design as a social process of information transformation, the systematic analysis of information flows becomes relevant. The perceived importance of these information flows is manifest in the number of studies in engineering design treating the subject or making frequent use of it (e.g. Steward 1981; Yassine, Falkenburg, et al. 1999; Campos Silva et al. 2012; Pektaş & Pultar 2006).

In generic terms, **information transformation** (or information processing) is the activity by which meaning is assigned to information inputs, subsequently transforming those information inputs into knowledge (Bruce and Cooper, 2000). More specifically, in the context of design activities, information transformation allows abstract statements of requirements to become detailed specifications of an engineering system, usually in the form of graphic and textual representations (Chira, 2005; Culley, 2014; Hubka, 1996; Shears, 1971). As a result of this information transformation, the work performed at each design activity generates and transforms exchangeable information packages that later constitute information flows between activities.

Although the expression 'information transformation' is not as common as 'information processing', the term has been more consistently associated with the idea of a collective or social process by which information is transformed or processed (e.g. Sim & Duffy 2003; Hubka et al. 1988; Durugbo 2015; Lindemann 2003, pp.105–110). In contrast, information processing has been applied more widely to the micro-cognitive (e.g. Alexiou et al. 2009), individual (e.g. Turner & Makhija 2012; Simon 1979), and organisational and inter-organisational levels (e.g. Premkumar et al. 2005). Although this thesis draws on information processing literature, I use the term 'information transformation' to capture the social aspect of information processing in the context of a design activity.

Small-scale design processes concentrate a big part of the information transformation (or processing work) in one or only a few designers, but when design processes involve a large number of participants and interdependent activities, a collective process of information transformation is required to coordinate actions and define the design object. In such larger processes, one designer's information outputs might be straightforward inputs for another designer, a situation that has been called information handover activities or 'over-the-wall' design (Eckert and Stacey, 2001). In other situations, the actual process of information transformation becomes a more intense social process, a collective process of negotiation and argumentation by which an integration of distributed design efforts occurs (Boujut and Laureillard, 2002).

Despite the social considerations mentioned, the information interdependencies between design tasks guide, implicitly or explicitly, the design process (e.g. Yassine, Falkenburg, et al. 1999; Danilovic & Browning 2007). These information interdependencies between parameters set the requirements for the information transformation that should occur within each task and between information-dependent tasks (Clarkson and Hamilton, 2000; Wynn et al., 2006).

2.1.2 Design as a technical process

In order to solve a technical problem, we need a system with a clear and easily reproduced relationship between [tasks'] inputs and outputs… Such relationships must always be planned–that is, designed to meet a specification.

(Pahl et al., 2007, p. 31)

Because the design of engineering systems is increasingly large in scale and technically complex, a significant amount of research efforts have been concentrated on mapping the design process as a set of interdependent technical tasks occurring over time (e.g. Ulrich & Eppinger 2012; Eppinger & Salminen 2001). This approach has provided the means to improve planning and analysis of large design processes, optimise task sequences, decompose or group tasks more efficiently, and evaluate critical tasks or process bottlenecks (Browning and Ramasesh, 2007; Eppinger and Browning, 2012). All these have contributed to a better understanding and management of the inherent complexity of large task networks.

This perspective of the design process is closely related to the design object and its architecture, as each task can be mapped to a specific component, subsystem, or to the integration and management of information required to design the engineering system. Consequently, the 'technical' design process is often organised based on the architecture of the design object, and vice versa (e.g. Yassine & Wissmann 2007; Sosa 2000). The assumption is that more alignment between their architectures will lead to increased efficiencies, especially for the design of well-known engineering systems. However, and as will be discussed in section 2.6, modelling the design process based primarily on technical aspects (such as task interdependency) can lead to an underestimation of the role of social processes in information transformation. For instance, a disregard for the social process perspective can generate a disconnection between task interdependencies and actual information flows. Such disconnection is problematic because those information flows are what ultimately allow addressing information dependencies in the first place.

2.1.3 Design as a socio-technical process

Design methodology now has to address the design process as an integration of all three of these: as a technical process, as a cognitive process and as a social process.

(Cross and Cross, 1995)

The view of the design process as a complex socio-technical process and a system of information transformation has a consistent, albeit fragmented theoretical grounding. It is hard to argue that a holistic understanding of the design process can exist without addressing technical and social dimensions, yet practical limitations have made such integration a challenging task. One reason for such practical difficulties lies in the different system definitions and boundaries utilised to describe and analyse the design process.

A common definition of system is *'a set of interrelated components working together toward some common objective.'* (Kossiakoff et al., 2011, p. 3). One reason for some of the practical difficulties in integrating social and technical dimensions is that the engineering design process as a system has at least three interdependent domains: the product, the organisation, and the process domains (Browning, 2001; Eppinger and Browning, 2012), each a system in its own right. These three domains also have their own architectures describing their elements and the interactions between them. Additionally, as the domains are interdependent, interactions between elements in different domains can be mapped, revealing valuable insights (Lindemann et al., 2009; Maurer, 2007).

In the product domain, we find components (grouped in subsystems if they exist) with interactions between them that can be material, spatial, of energy flow, of information flow, and so on (Eppinger and Browning, 2012, p. 18)

In the organisation domain, we find people and their interactions, which in the context of the engineering design process are usually related to communication, and more specifically, information exchanges (Steward, 1981; Yassine, Falkenburg, et al., 1999). The analysis in this domain can be at the level of people or aggregated into groups, departments, or organisations.

In the process domain, the architecture can be described in terms of the engineering system that is being designed, focusing on information dependencies between design tasks (e.g. Collins et al. 2009; Eppinger et al. 1994). Alternatively, the architecture can be described in terms of the organisation that performs the design, focusing on information flows between design activities (e.g. Morelli et al. 1995; Parraguez et al. 2014). Unless explicitly addressed, this mixed nature of the process domain, in addition to the distinctions between the planned and the actual architectures, can become a source of conceptual confusions and analytical limitations.

In order to move towards a characterisation of the design process as a socio-technical system of information transformation, the utilised model must consider these nuances. At the conceptual level, such an integration requires actionable and interconnected models of the design process that do not add to the complexity (Browning, 2009). At the empirical level, models should be testable using real-world data (Morris, 1967), something proven to be difficult, even when only one aspect (social or technical) is addressed (Smith and Morrow, 1999).

2.1.4 Models and modelling of the design process

As several studies have shown (Maier et al., 2013; Roozenburg and Cross, 1991; Wynn, 2007), models of the design process, as well as the act of modelling the design process, can be effective tools for improving the process, especially for large engineering design projects where the design process is far from self-evident. Three fundamental questions must be considered before further elaboration:

- What is a model?
- How do we 'model' a model?
- What is a process?

After combining the answers to these questions with the discussion about the design process previously presented, we can examine different design process models, explore issues related to design process modelling, and if required, develop alternative models.

What is a model?

The word 'model' has different meanings and can be used as either a noun or a verb. In this thesis, the noun 'model' is defined as: '*an approximation, representation, or idealization of selected aspects of the structure, behavior, operation, or other characteristics of a real-world process, concept, or system'* (IEEE Standards Board, 1989, p. 12), that is, an abstraction.

In addition, a model might be instantiated through different views depending on the specific purpose (Browning, 2009). Each model view is understood as a representation of a system from the perspective of specific concerns or issues (IEEE Standards Board, 2000, p. 3).

Therefore, a model should help us to understand a particular phenomenon or object through meticulous reduction of its overall complexity. The simplification should be enough to enable an analysis of the phenomenon or object while maintaining as many useful details as possible.

Models can be classified by the degree of detail they include and their objective (Wynn, 2007). Models may be very general, like Cross's model of the design process 'Exploration →

Generation → Evaluation → Communication' (Cross, 2000), or very detailed, like those that describe interactions among product components (e.g. Sharman & Yassine 2004).

As for their objective, a model may be descriptive, prescriptive, or both (Blessing, 1994, p. 13; Heisig et al., 2010, chap. 1; Wynn, 2007). A descriptive model attempts to capture reality 'as is', in order to increase current understanding of the phenomena or object under study. For example, a descriptive model may enable exploration of causal relationships between the architecture of a system and its behaviour. In contrast, a prescriptive model attempts to portray things as they should be, based on perceived best practices. Prescriptive models often derive their recommendations from insights obtained through descriptive models that imply causal relationships or through models that explain the mechanisms that connect independent and dependent variables, such as performance (Heisig et al., 2010, chap. 1; O'Donnell and Duffy, 2002). Figure 2-3 summarises the previous consideration using a model classification matrix suggested by this thesis.

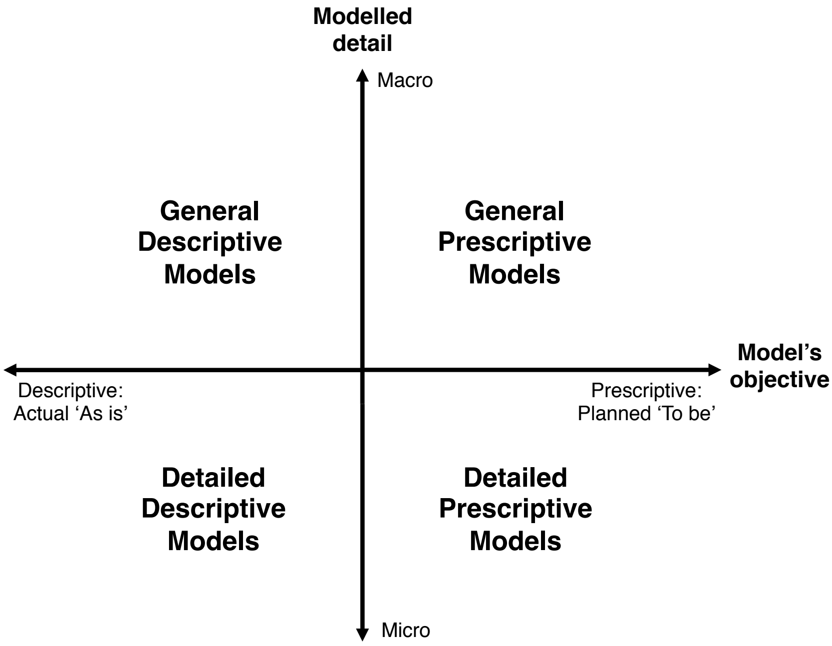


Figure 2-3: Model classification matrix

How do we 'model' a model?

Consistent with the previous definition, this thesis defines the verb 'model' (to model or modelling) as the act of devising a representation (model) of a phenomenon or object. In other words, 'modelling is a process of abstraction from the *real* world' (Smith and Morrow, 1999). Depending on the objective and the detail of *the model that is being modelled*, the way in which modelling occurs changes. To create a new descriptive model requires empirical data directly from the modelled phenomenon and/or inputs from previous descriptive models (Cross, 2000, p. 29). In turn, to create a prescriptive model requires either first developing a descriptive model (to obtain the causal relationships or mechanisms that sustain the prescriptions) or using previous models and elaborating on their findings. Typically, general models can be based on more detailed models, but not the other way around. Figure 2-4 illustrates the previously described movements using the model classification matrix.

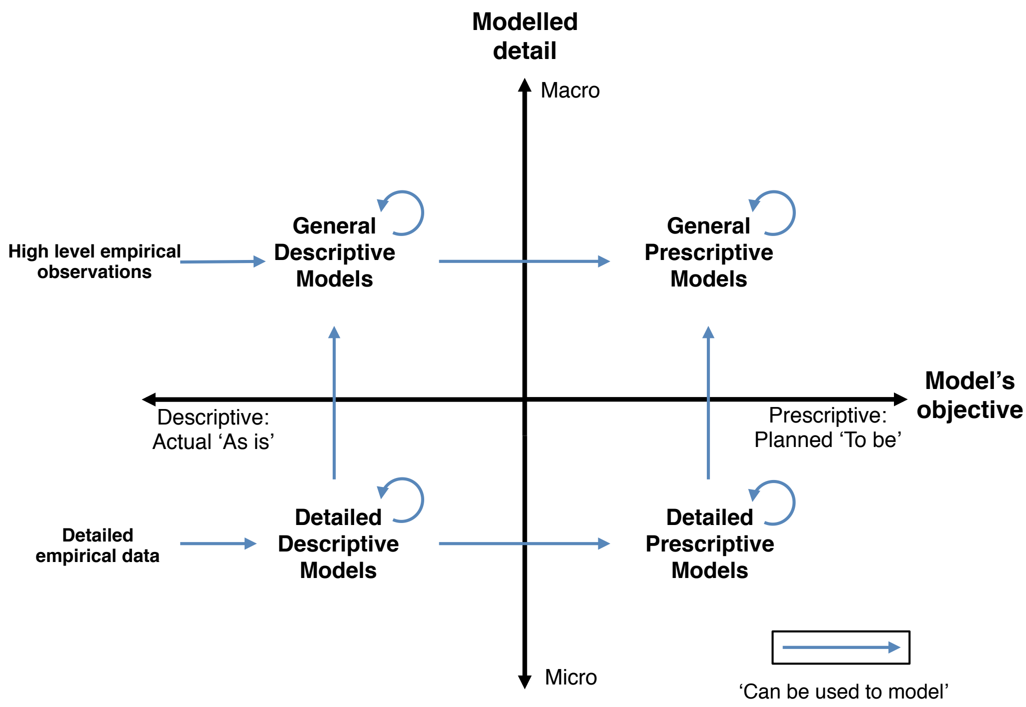


Figure 2-4: Diagram of main movements in the model classification matrix

What is a process?

In the context of this thesis, a process is considered to be 'a series of actions or steps taken in order to achieve a particular end' (Stevenson, 2010). A key characteristic of a process is that, unlike people in an organisation or product components, it does not exist as a clear physical entity, but only as a construct, an idea containing various elements. For example, human processes such as designing are constituted by a set of interconnected actions (verbs), things towards which the actions are directed (objects), and 'doers' (subjects, or those who perform the actions).

The non-material and temporal nature of processes makes them an active area for modelling. In fact, only through models can we describe process characteristics and prescribe actions for improvements (Buede, 2009, p. 73); therefore, models and modelling have paramount importance in the design process of complex systems.

Design process models

With these generic definitions of 'design', 'process', 'models' and 'modelling', it is now possible to develop a working definition for **design process models within the scope of this thesis**:

A model of the design process is a representation of a system of information transformation, a system with the objective of transforming a set of requirements into a detailed design. **Models of the *actual* design process** represent the process 'as is' in a primarily **descriptive** manner; in contrast, **models of the *planned* design process** represent the process as it is 'to be' in a primarily **prescriptive** manner.

This broad definition permits a discussion of the design process as a complex socio-technical system of information transformation, while simultaneously encompassing descriptive and prescriptive models at multiple levels of abstraction. Figure 2-5 illustrates the previous points using examples of process models in the four quadrants of the matrix. In this new matrix the Y axis has been relabeled to better reflect the specific meaning of 'micro' and 'macro' modelled detail in the context of design processes.

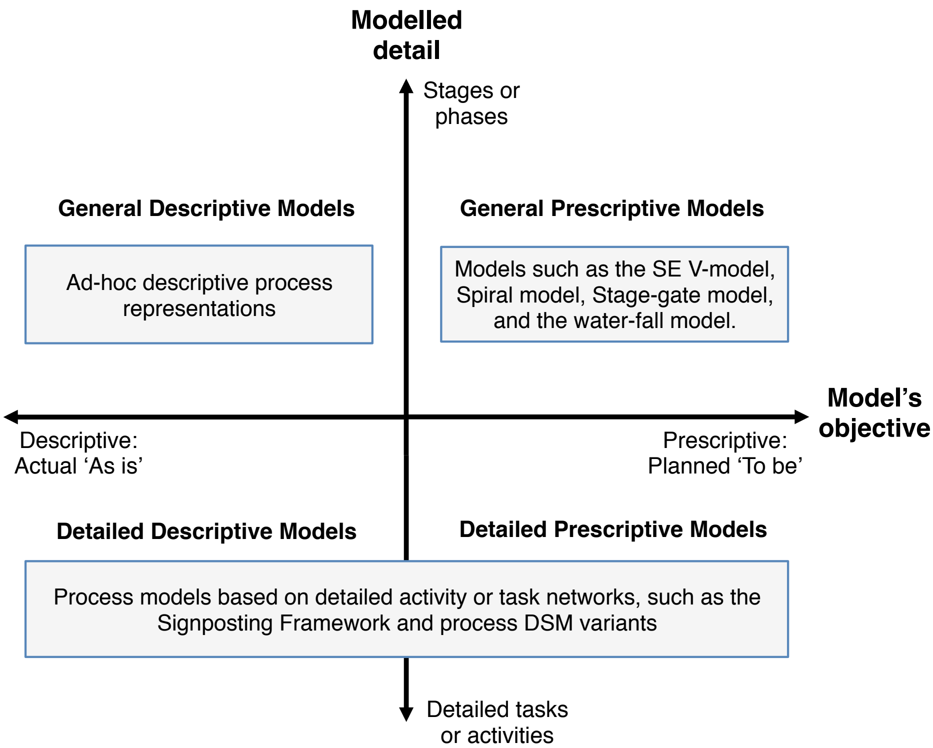


Figure 2-5: Process model examples mapped into the model classification matrix

The two lower quadrants in figure 2-5 concentrate process models based on tasks or activities, and their higher level of detail allows for the application of analytical methods. Depending on the information gathered and assumptions made, these models can be descriptive, prescriptive, or a mixture. This category includes activity and task network models, such as DSM-based process models (Eppinger and Browning, 2012; Steward, 1981) and the Signposting Framework (Clarkson and Hamilton, 2000; Wynn et al., 2006).

The top-left quadrant concentrates general and descriptive process models, defined by a macro view of the process and a contextualised description of a specific industrial model. The top-right quadrant concentrates general prescriptive models, which also are defined by a macro view. This category includes models such as INCOSE's System Engineering V-Model (Haskins et al., 2011), Cooper's Stage-Gate model (Cooper, 1990), the Spiral Process Model (Boehm, 2000), and other stage-based variants of process models.

As illustrated in figure 2-4, it is typically possible to derive general process models from those that provide a higher level of detail. Therefore, this review focuses on detailed process models based on task or activity networks, which, if needed, can be used to build more general descriptive or prescriptive models that aggregate tasks or activities into stages or other process views.

2.2 Complexity and applied network science

According to previous research (Cilliers, 2000; Holland, 1997; Johnson, 2007; de Weck et al., 2011), a system is complex if it fulfils all of the following: 1) contains multiple elements, 2) possesses a number of connections between the elements, 3) exhibits dynamic interactions among the elements, and 4) exhibits emergent behaviour (the interactions among the elements produces behaviour that cannot be explained by the simple sum of the elements). In addition, although not a requirement for complexity, the interconnectedness among the system's elements should be higher than the interconnectedness between the system and its environment. Simon (1962) has termed this quality 'near decomposability', and it serves to distinguish the system from its environment and to identify meaningful levels of analysis.

Although a complex system might also be considered complicated, it does not need to be. More precisely, something complex might be perceived as complicated simply because of an insufficient overview, inadequate representation (Simon, 1962), or a lack of understanding of the driving forces. With the right conceptual and analytical tools, the perception of complicatedness can disappear while complexity remains. Conversely, something complicated or intricate is not necessarily complex. The mere existence of multiple interconnected elements does not necessarily translate into emergent behaviour (Johnson, 2007). A borderline situation occurs in systems labelled as chaotic. In such systems, although emergent behaviour does occur and all the previous conditions are met, the behaviour is so unpredictable that it appears random. The absence of distinguishable patterns makes chaotic systems inherently complicated and a challenge for traditional modelling techniques applied to complex system (Sheard and Mostashari, 2009). A premise in this thesis is that although the design process can exhibit some chaotic behaviour, the system as a whole, particularly its architecture, is predominantly complex, not chaotic.

Based on this understanding of complexity, a number of biological, social, technical, and socio-technical systems have been conceptualised and successfully analysed as complex systems, despite differences in their scales and nature (Barabási, 2002; Newman, 2003). The rise of complexity science as a tool to study such diverse systems is partially attributable to the theoretical and analytical foundations of this new science, which facilitates the understanding of emergent behaviours. Moreover, such emergent behaviours are what generate and allow us to explain 'higher-order' properties and functions, like self-reproduction, self-organisation, intelligence, and communication (Alexiou, 2010; Holland, 1997). Consequently, understanding how a complex structure of interactions can generate useful (or harmful) behaviours is crucial to improve the design, production, and management of human-made engineering systems (Calvano and John, 2004; Storga et al., 2013).

If complexity is broadly about a number of **elements** and their **interactions**, we can distinguish two facets that are fundamental in any effort to analyse and manage complexity: **structure** and **composition** (Wasserman and Faust, 1994). **Structure** can be generically operationalised as the arrangement of the elements of a system and their inter-relationships. **Composition** can be operationalised as the combination of elements of a system. In other words, through a system's structure we obtain how things connect; through a system's composition we obtain information about the nature of the various connected elements. With the combination of structure and composition, we can capture some of the key building blocks of complexity and investigate how a system's architecture affects the system's behaviour.

The design activity of even one or only a few individuals is in its own right complex because of the difficult cognitive task of dealing with ill-defined and sometimes conflictive requirements, tight constraints, and changing environments (Cross, 2004). With ever-larger and technologically challenging projects, the field of Engineering Design has been increasingly exposed to higher levels of social and technical complexity (Bartolomei et al., 2012). As a result, the use of complexity-based approaches has increased along with the need for a deeper understanding of the source of emergent behaviours arising from the design object and process.

To analyse the effects of the design process's architecture on the performance of complex engineering design processes, we need not only a model but also a quantitative method. Because of the characteristics of complex systems, and the distinction between structure and composition, network-based approaches have been widely applied means to quantify architectures (Baldwin et al., 2013; Browning and Ramasesh, 2007). In addition, statistical network analysis has shown that certain network properties are common to a range of different systems (Albert and Barabási, 2002; Braha and Bar-Yam, 2007). Therefore, these properties and their measures can be useful parameters to analyse and interpret network structures found in complex engineering design projects.

2.2.1 Network-based methods to analyse complexity

A system of interconnected elements can be modelled through its network architecture. As with complexity in general, network architecture can be examined in terms of structure and composition (Phelps, 2010; Wasserman and Faust, 1989, 1994). The network structure provides a quantitative and/or graphical representation of the 'interconnectedness' among the elements, while the network composition describes the characteristics of the network's constituent elements and quantifies the diversity of those attributes. The network's elements are commonly referred to as nodes, vertices, or points, and the relationship between two nodes is commonly represented as a line and referred to as edges, links, ties, or arcs (Wasserman and Faust, 1994, p. 95). Depending on the particular network method and available data, a network model's richness and analytical possibilities can vary widely.

As a minimum, a network-based approach will consider a list of elements and make a binary indication about the existence or non-existence of a relationship between each element. Despite this elementary way of modelling a complex system, even basic network models allow us to gain insights about a system's key features and properties. For example, through network-based approaches, we can measure interdependence and decomposability and describe modularity in systems (e.g. Fixson 2007; Browning 2001). Moreover, there is also evidence of recurrent network topologies and evolutionary trends in natural and technical systems, a recurrence that seems to be based on robust and common organising principles (Albert and Barabási, 2002).

In the case of large networks composed of thousands or millions of elements, such as regional communication systems, energy grids, and transport networks, a simplified characterisation of network structure and composition can provide sufficient information to study the impact of the network architecture on the system behaviour. However, smaller networks, such as those describing design processes, require higher levels of detail because their complexity lies proportionally less in the overall network structure (also called topology) and more in the local structural and compositional characteristics (Carrington et al., 2005). Therefore, to study design process networks, we must consider structural and compositional network characteristics in additional detail in order to capture more accurately the complexity of the system (Wasserman and Faust, 1989).

Network architecture: structure and composition

Structure: In addition to whether a relationship exists between any two elements, the **network structure** can have the following features:

- **Relationship type(s):** A system can be modelled as a network based on various types of relationships between its elements. For example, a relationship between two elements may be characterised as energy flow, information flow, material exchange, spatial, and so on. A relationship also may be planned, one that describes an expected information dependency, or actual, one that describes a previous or current interaction. In addition, a system can be modelled that combines various relationship types. Depending on the modelling approach, each type of relationship may be analysed as a separate network layer, known as a multiplex approach, or combined into a unique a network, following variants of multi-modal networks (Kivelä et al., 2013). It is important to notice that the combination of different relationship types into one network, although computationally possible, can represent interpretational challenges if the relationship types are conceptually incompatible. Therefore, multiplex or multi-modal models should explicitly account for these interpretational challenges. (D'Agostino and Scala, 2014, chap. 2)

- **Edge weight:** A relationship between two nodes can be valued based on aspects such as its relative strength, impact, and/or frequency. Weighting the edge between two nodes provides the means to distinguish different degrees of connectivity, even if the network is fully connected. This ability is particularly important because in practice, especially in small and dense networks, the structure of the network is determined by the intensity of the relationships between elements (Wasserman and Faust, 1994, p. 140).
- **Relationship directionality:** A relationship between two elements may or may not be directed; that is, the energy or materials usually flow in a particular direction. Likewise, information exchange may or may not be reciprocated. As a result, networks are classified as directed or undirected. For a network to be directed, one or more of the relationships must occur in only one direction. (Wasserman and Faust, 1994, p. 121).
- **Dynamics:** A network structure may change over time because of the following reasons: a) nodes appear and/or disappear, b) edges appear and/or disappear, and c) edges change direction, are reweighted, and/or change relationship type. When analysing a system's structure over long time periods, the dynamic evolution of the system can be the key to understanding its structure. This is particularly true when studying complex and evolving processes such as design, where changes in the emergent behaviours are not only expected but needed to fulfil envisioned objectives (Niloy et al., 2009).

Composition: In addition to a simple list identifying each node, **network composition** can have the following features (Wasserman and Faust, 1994):

1. **Node attributes:** A node represents an element or part of the system. As a minimum, each node must have at least one attribute, a unique ID. In addition, it can have other

not necessarily unique categorical and numerical attributes providing supplementary information about each node (Wasserman and Faust, 1989).

2. **Node types:** A network model can contain more than one type of node, and the number of node types is referred to the network's modality. One-mode networks contain only one type of nodes, two-mode networks contain two, and so on. (Wasserman and Faust, 1994). A combination of nodes types may generate a combination of relationship types, and consequently, additional interpretational challenges. However, depending on the theoretical and practical requirements, a combination of node types could be necessary to better reflect the modelled system's actual composition. For the network model to have internal consistency, a node generally can be associated with only one type at any given point in time. In contrast, a node may have multiple attributes.

3. **Nestedness**: In a non-nested network, each element or part cannot be further decomposed as another network. In contrast, in a nested network, one or more nodes may be modelled as a 'container' that can be decomposed into another network. When a network model is nested, the relationship between a decomposable node and any other node in reality will contain a bundle of relationships, which summarises all the relationships between the examined nodes. Such a hierarchical nestedness is a network operationalisation of the concept of systems-of-systems (Clark, 2008). Nested network models formalise inherent modularities and facilitate the analysis and interpretation of networks that otherwise would contain too many heterogeneous elements and relationship types. Although nestedness also could be a structural characteristic of a network, here it is as a compositional one because once nestedness is applied to a network model, it becomes a feature of a node that allows complexity to be encapsulated.

4. **Dynamics:** Both the structure and the composition of a network can evolve through dynamic changes in the nodes' attributes. For example, people can change affiliations, or the risk associated with an activity can vary over time. Depending on the type of system modelled and the time frame under analysis, the effect of these dynamic changes on the system's behaviour can be significant (Holme and Saramäki, 2012).

As previously discussed, **the combination of compositional and structural network characteristics is what is understood as 'network architecture' in this thesis.** The premise here is that network architecture is at the heart of emergent behaviours, and therefore, understanding the architecture can lead us to anticipate and design behaviours that better match the intended functions (Figure 2-6). This objective is consistent with John Gero's Function-Behaviour-Structure (FBS) ontology (Gero and Kannengiesser, 2004); however, this thesis distinguishes between structure and composition, whereas, as Dorst and Vermaas (2005) pointed out, in the FBS ontology the idea of composition is contained by the definition of structure:

By the structure of an artefact is meant the materials its components consist of, the dimensions of these materials and components, and the way these materials and components are related geometrically (Dorst and Vermaas, 2005, p. 19).

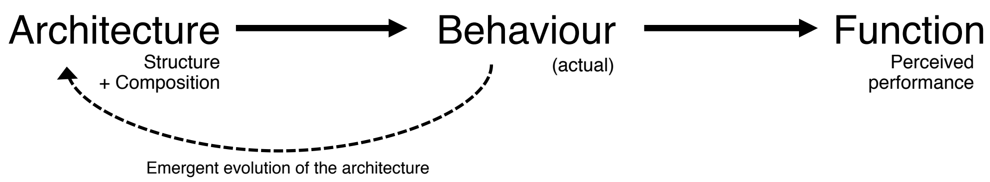


Figure 2-6: The relationship between architecture, behaviour, and function. Adapted from Gero and Kannengiesser (2004), Function-Behaviour-Structure framework

Network architecture: levels of analysis

By its own nature, the study of a system through network-based models and methods is inherently multilevel (Iordache, 2011, p. 2). Three fundamental levels common to any network representation are the node, the edge, and the whole network level (Wasserman and Faust, 1994).

- **Node level:** The analysis at this level focuses on each element of a system and can include the effects of the immediate neighbours to which the focal node is connected. In social networks, this level is often referred to as 'ego-centric' or 'ego-network'. For nested networks, an analysis at the node level implicitly or explicitly examines the whole network nested inside the focal node. For example, a node-level analysis of actual process architecture would allow focusing on each activity.
- **Edge level:** Analysis at the edge level focuses on each relationship between any two elements of a system, and depending on the decision of the modeller, may include elements at each side of the edge. Contextualised to product and organisation architectures, this level is often referred to as 'interfaces' (e.g. Sosa et al. 2004), and in social networks is sometimes called 'dyadic relationships' (Carpenter et al., 2012). For nested networks, an edge-level analysis involves all relationships between the nested nodes connected to the focal edges. For process architecture, an analysis at the level of each edge typically focuses on identified information dependencies between tasks or on actual process interfaces between activities.
- **Whole network level:** At this level, the analysis includes the entire network of nodes and edges, and for a process, provides a characterisation for the whole structure and composition that describes the process under study.

Figure 2-7 shows a visual representation of the three levels and their relationships with each other. While nodes and edges are hierarchically at the same level, I discuss and analyse them separately in this thesis because both are further decomposed in the developed framework. Though such decomposition they become nested networks, and as such, each has its own network structure and composition.

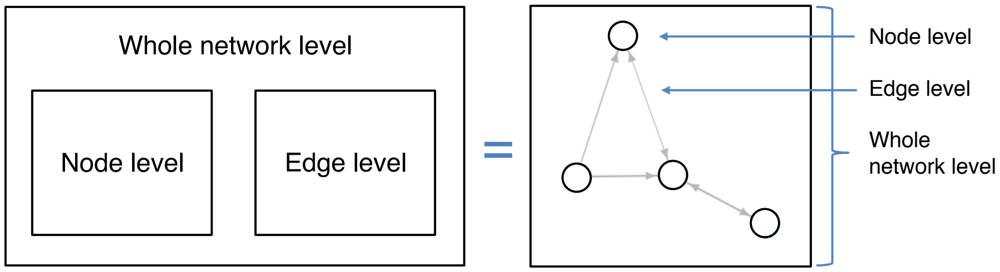


Figure 2-7: Diagrammatic representation of the three network levels

Having established the key concepts of complexity and the building blocks of network-based methods used to analyse complexity, I introduce the most common network concepts applicable to engineering design in the next section.

2.2.2 Applied Network Science and Engineering Design

The several models and methods for analysing complexity that can be described as 'network-based' differ in how they represent the network and the degree of analytical detail they can achieve. The most common approaches for engineering design and engineering systems are either **matrix-based** or **graph-based**. The most obvious difference between these two is that matrix-based approaches use square or rectangular matrices and graph-based use network graphs. This representational difference is rooted in different analytical methods, needs, and assumptions (Wyatt et al., 2013), each of which brings distinctive strengths and weaknesses. For instance, matrix-based approaches often have been associated with optimisation methods such as sequencing, clustering, and partitioning (Browning, 2001). In contrast, graph-based approaches can be used to describe and analyse network structures at different levels with a wide variety of specific metrics, such as centrality and density (Wasserman and Faust, 1994). To leverage the strengths of each approach, recent studies have combined matrix-based and graph-based analyses (e.g. Pasqual & de Weck 2011; Collins et al. 2009).

Some of the most widely utilised matrix-based approaches in Engineering Design and Systems Engineering are the Design Structure Matrix (DSM) and its variants, the Multi-Domain Matrix (MDM), and the Domain-Mapping Matrix (DMM) (Maurer, 2007). DSM is a flexible method based on square matrices (also known as influence or adjacency matrices) that makes explicit the connections between two elements of the same domain (Eppinger and Browning, 2012; Steward, 1981). Traditionally, DSM has been used to focus on the same three domains previously discussed: product, organisation, and process architecture. Product architecture DSM analyses dependencies/interactions between components; organisation architecture DSM analyses communication/interactions between people; and process architecture DSM looks at dependencies and information flows between activities. In addition to DSM, the Multi-Domain Matrix (MDM) allows for mapping connections between domains, such as mapping organisation to process, process to product, and so on. In turn, each cross-domain mapping can be represented by the Domain Mapping Matrix (DMM), a rectangular matrix that maps elements from one domain to the other (Eppinger and Browning, 2012; Maurer, 2007), for example, connecting people to activities. Yassine, Whitney, et al. (2003) earlier introduced such cross-domain mapping as connectivity maps.

In contrast to matrix-based approaches, graph-based approaches are more diverse, varying widely in terms of analytical capabilities and focus. Nonetheless, all graph-based approaches share a representation based on nodes and edges, sometimes referred to as 'boxes and arrows' (e.g. Eppinger 2001; Kreimeyer & Lindemann 2011). On one side, simpler graph-based approaches do not have a quantitative intent, but instead, provide only a graphical summary of architectural information. Examples include organisational charts, workflow diagrams, and basic abstract representations of a product's architecture. On the other side, approaches such as Petri-Nets, variants of social network analysis, IDEF0 and IDEF3 diagrams, and PERT and GERT diagrams are not only intended to visualise but also analyse the network at one or more levels (Browning and Ramasesh, 2007).

A generic and therefore flexible graph-based approach is the direct application of graph theory to analyse a network. Given the socio-technical nature of the design process, one common variant of graph theory that researchers have applied in the field of engineering design is social network analysis (SNA) (Wasserman and Faust, 1994). Despite its name, SNA methods have the flexibility and extensive set of metrics at multiple levels of analysis[2] to be used to study not only social architectures, but also process (e.g. Collins et al. 2009) and product (e.g. Sosa et al. 2007a) architectures.

2.2.3 Quantifying networks

To analyse systematically the impact of network complexity on a given system, we must first quantify at the same level the independent and dependent variables that will be used as the base for study. In this case, the independent variables are metrics of network structure and

[2] These levels include analysis of whole networks and network subsets, 'ego-network' analysis of each node in terms of its network neighbourhood and network embeddedness, and edge analysis via linegraphs and other transformations.

composition, while the dependent variables are metrics representing performance or any other behaviour of interest. This subsection centres on the former, metrics that characterise the architecture of systems.

The development of network metrics to quantify a system's architectures has been an active, yet fragmented area of research (Kreimeyer and Lindemann, 2011). No unique, consistent body of metrics with standardised naming conventions and a clear taxonomy can be found to encompass the wide range of application domains and disciplines involved (Estrada et al., 2010), stretching across fields as diverse as Biology (systems biology), Physics (complex physical systems), Sociology (social networks), and Engineering (engineering systems).

Based on the scope of this thesis and existing comprehensive reviews of a range of network-based complexity metrics (e.g. Kreimeyer & Lindemann 2011, chap.2; Barabási 2012, chap.2), my emphasis here is mostly on the network metrics more directly applicable to this study. I selected these metrics based on three criteria: 1) the ability to describe the most fundamental characteristics of a network's structure and composition at each level of analysis, 2) the ease of interpretation, and 3) the existence of low correlation between each of the selected metrics, or in other words, the metrics should represent a distinct non-overlapping network characteristic.

The selected metrics for network structure and composition are introduced below.

Quantifying Structure

Structural aspects describe the topological characteristics of a network's architecture, that is, the particular configuration of connections between elements. Metrics that allow these aspects to be quantified include those that measure structural characteristics for each node, for the whole network, and for the edges.

At the level of each node, the network structure surrounding the node is measured, or in other words, the location of the node in the network. The assumption is that the degree of the node's embeddedness in the whole network affects its potential impact as well as the potential impact of the whole network in the node (Bonacich, 1987; Wasserman and Faust, 1994). A family of metrics known as node or point centrality captures the core element of each node's structural characterisation (Freeman, 1979). Node centrality metrics quantify the connectedness of each node and are often interpreted in relative terms by comparing them against the measures of the other nodes in the network. Such centrality metrics and their variants are the most frequently utilised measures to capture structural characteristics at the node level

(Abraham, 2010, p. 29; Wasserman and Faust, 1994, p. 169). Some of the most widely adopted node centrality metrics and their definitions are summarised below:

- **Degree centrality** measures the number of edges (connecting to other nodes) that a node has. In a directional network, it is possible to distinguish between in-degree (number of incoming edges) and out-degree (number of outgoing edges) (Wasserman and Faust, 1994).
- **Closeness centrality** captures the closeness of the node to all other nodes in the graph. More specifically, this metrics is defined as '…the sum of graph-theoretic distances from all other nodes, where the distance from a node to another is defined as the length (in edges) of the shortest path from one to the other' (Borgatti, 2005).
- **Betweenness centrality** captures in essence 'the number of times a node acts as a bridge along the shortest path between two other nodes' (Freeman, 1977). More specifically, this metric is defined as '… the share of times that a node *i* needs a node *k* (whose centrality is being measured) in order to reach a node *j* via the shortest path' (Borgatti, 2005).
- **Eigenvector centrality** is a measure in which the centrality of a node depends on the centrality of its direct neighbours; in other words, the more central a node's connecting nodes are, the more central is that node. Mathematically, 'eigenvector centrality is defined as the principal eigenvector of the adjacency matrix defining the network' (Borgatti, 2005).
- Stephenson and Zelen's (1989) **node information centrality** is not frequently used, but is one of the few centrality metrics that can be applied simultaneously to weighted networks and that considers all paths between two nodes, not just the shortest distances (Wasserman and Faust, 1994, p. 197). As such, node information centrality generalizes the centralities of closeness and Freeman's betweenness. This metric is particularly useful for the study of networks with undirected or reciprocated information exchanges, where the objective is not to find the shortest path, but to identify and weigh all possible routes by which information can be exchanged.

At the whole-network level, the overall network structure or topology describing the sum of connections among elements is measured, as well as the number of elements in the network. The assumption is that the network's overall structure affects the characteristics of the system described (Wasserman and Faust, 1994, p. 112). This effect can be materialised in aspects such as the system's resilience when facing environmental disturbance and its capacity to deal efficiently with information flows. Some of the most widely adopted metrics for whole networks[3] (Faust, 2006) and their definitions are summarised below:

- **Network size**: Number of elements in the network.
- **Network density**: Number of actual edges divided by maximum potential amount of edges.
- **Centralisation**: Freeman (1979) offered a widely accepted group measure of betweenness centrality. This measure ranges from 0 to 1 and is computed by considering the betweenness centralities of all nodes. The closer an actor is to 1 '… the more likely it is that a single actor is quite central, with the remaining actors considerably less central…. this group-level quantity is an index of *centralization,* and measures how variable or heterogeneous the actor centralities are. It records the extent to which a single actor has high centrality, and the others, low centrality.'

Network metrics and research **at the level of each edge** are comparatively less abundant; however, some quantitative methods can be applied to capture structural characteristics at this level, too.

- **Multidimensional Scaling (MDS)** uses relational data to express the closeness or similarity between each pair of nodes, thus, describing a structural value for each edge (Scott, 2000, pp. 148–153).
- **The information centrality nearness matrix** also captures the closeness or nearness between two nodes to characterise an edge. This nearness matrix is produced as a middle step in the calculation of the node-level information centrality measure (Stephenson and Zelen, 1989). In the nearness matrix, 'the distances are converted to "closeness" by taking reciprocals, and a closeness measure is constructed by taking the harmonic mean of each row of the nearness matrix' (Borgatti and Everett, 2006, p. 473).
- **The line graph** is a theoretical construct and computational method that can convert edges into nodes and vice versa. The line graph describes 'which relations in the graph are adjacent to which other relations. Two relations are adjacent if they share an actor.' (Hanneman and Riddle, 2005). With this method, any node-level structural network metric can be calculated in terms of edges. Although the line graph has been largely a theoretical approach, recent research has shown its usefulness in areas such as graph community

[3] Any network subset containing at least two nodes and one edge can be examined as a network.

detection (Fortunato, 2010). For this thesis, the line graph provides a useful tool to maintain consistency and completeness across the three levels of analysis, transforming an edge into an equivalent node representation.

Quantifying composition

Compositional aspects describe the type and variety of elements (nodes) found in the studied network, which includes nominal attributes such as the numbers of departmental affiliations, demographic characteristics, or any other relevant feature of the network nodes.

Although superficially the quantification of compositional diversity might appear to be straightforward, in reality measuring composition based on the diversity of nominal attributes goes beyond simply listing elements and counting the various types (Magurran, 1988). The challenge is to boil down compositional diversity into one metric that is consistent, easy to interpret, and comparable across different networks. Additionally, such a metric needs to meet the previously mentioned challenges regardless of the amount of elements in the system, the actual amount of element types, and the maximum potential amount of element types.

An academic field with long experience in measuring compositional diversity is the field of ecology, where measures such as Shannon's, Brillouin's, and Simpson's diversity indices have been developed (Magurran, 1988). One compositional diversity index with all the required attributes that stands out as a robust and transparent method to quantify compositional diversity is the **Index of Qualitative Variation (IQV)** (Agresti and Agresti, 1977; Frankfort-Nachmias and Leon-Guerrero, 2011; Verma, 2012, p. 46). IQV has been used in studies ranging from ecological diversity to social network analysis to calculate the relative heterogeneity of a network in terms of the variety attributes in the network's population (Halgin and Borgatti, 2012). This index is a normalised and continuous measure from 0 to 1 in which 0 means no heterogeneity (all participants come from the same functional group) and 1 means maximum heterogeneity (each participant comes from a different functional group). The IQV index is calculated as follows:

$$IQV = \frac{K(100^2 - \sum Pct^2)}{100^2(K-1)}$$

where

K = the number of categories (for example, the total number of departments), and

$\sum Pct^2$ = the sum of all square percentages in the distribution (as an integer number).

For example, the IQV index of a network of four members, only four departments, and each network member represents a different department (maximum heterogeneity) would be:

$$IQV = \frac{4(100^2 - (25^2 + 25^2 + 25^2 + 25^2)}{100^2(4-1)} = 1$$

Although IQV is not a metric found in previous engineering design studies, I found it to be the most suitable way to quantify compositional diversity because of its computational transparency, ease of interpretation, and flexibility to account for unlimited sample sizes and groups.

Dynamic network analysis

In addition to using these various metrics to calculate a purely static analysis based on one temporal snapshot, it is also possible to use the metrics to analyse the dynamic evolution of networks. However, dynamic network analysis is far from being a mature area, and there is no predefined form for calculating the evolution of each network metric (Holme and Saramäki, 2012). For example, the most frequent and simplest approach to analyse network dynamics is to divide the network into time segments (or snapshots), employ each metric as a static measure inside each segment, and then plot the results for each segment as discrete values over time. This method based on stacked snapshots has become a relatively standard practice, yet its simplicity hides important challenges and questions that lack clear answers (Boccaletti et al., 2006; Holme and Saramäki, 2012). Some of the most important considerations and open questions include:

- How large should the length (in time) of each segment or snapshot be?
- Should each segment accumulate the edges and/or nodes that appeared in the previous period(s) of time? If so, for how long should the previous nodes and edges remain?
- Should all nodes, edges, and attributes be dynamic, or should only some of them be?

These are relevant questions because the answers will modify the quantification of network characteristics. For example, decisions about the length of a segment and whether to accumulate nodes and edges from previous periods will affect the overall network size: On one extreme, too many segments (each accumulating a very short time period) could lead to periods with almost no activity or insufficient nodes to calculate meaningful metrics. On the other extreme, a few or only one segment risks hiding important information about the actual sequence of events and could lead to networks so large and densely connected that they are not representative of the real system (Boccaletti et al., 2006; Holme and Saramäki, 2012). For these reasons, an appropriate dynamic analysis demands a deep understanding of the system under study, based on a qualitative exploration of the effect of different analytical decisions, so that the researcher can break down the analysis into meaningful segments.

Table 2-1 summarises the network metrics for structure and composition, distinguishing structural metrics by level of analysis.[4]

Table 2-1: Examples of network metrics for structure and composition

Architectural dimension	Level	Metric	Description	Reference	Example of application in the literature
Structure	Whole network level	Size	Number of nodes	Wasserman & Faust (1994)	Jepsen (2013)
		Density	Relative connectedness of the network		Sosa (2008)
		Centralisation	Indicates the distribution of centrality in the network, i.e. it records the extent to which a single actor has high centrality, and the others, low centrality		Hossain et al. (2013)
	Node level	Degree centrality	Centrality measures describing the relative structural prominence of nodes to identify key elements in the network	Freeman (1979)	Collins et al. (2009)
		Closeness centrality			Batallas & Yassine (2006)
		Betweenness centrality			Leenders et al. (2007)
		Eigenvector centrality		Bonacich (1987)	Pappas & Wooldridge (2007)
		Information centrality		Stephenson & Zelen (1989)	Tortoriello et al. (2011)
	Edge level* *not actual metrics, but means to characterise edges quantitatively based on network structure information	Multidimensional Scaling	Method to measure the closeness or similarity between each pair of nodes	Kruskal & Wish (1978)	Oliver & Ebers (1998)
		Information centrality nearness matrix	Alternative method to measure the closeness or similarity between each pair of nodes based on the information centrality nearness matrix	Stephenson & Zelen (1989)	Borgatti & Everett (2006)* *Not an actual application, but a theoretical consideration.
		Line graph	Theoretical construct and computational method for turning edges into nodes and vice versa	Weisstein (2003, p.1776)	Fortunato (2010)
Composition	Whole network level	Ecological diversity indexes including Shannon's, Brillouin's, Blau's and Simpson's	Various measures of variability for nominal categories	Magurran (1988)	Chen & Gable (2013); Talke et al. (2011)
		Agresti's Index of Qualitative Variation (IQV)	Measure of variability for nominal categories	Agresti & Agresti (1977); Frankfort-Nachmias & Leon-Guerrero (2011, chap.5)	Halgin & Borgatti (2012); Borgatti et al. (2002)

[4] Appendix I provides a list of equations for the network metrics used in this thesis.

These structural and compositional metrics can be used to quantify any type of network architecture; however, the characterisation and interpretation of architectural characteristics require additional domain specific knowledge, some of which will be discussed in the following sections.

2.2.4 Domains and architectures

In the fields of Engineering Design and Engineering Systems, three domains are commonly recognized, explicitly or implicitly: the process, the organisational, and the product domains (Eppinger and Browning, 2012; Eppinger and Salminen, 2001). Each represents a distinctive area of knowledge and can be studied as a system with its own architecture and behaviour. In addition, other domains also have been proposed and analysed, such as the functional domain that contains objectives and functions and the environmental domain that contains the system drivers, like factors that act on the system or vice versa (Bartolomei et al., 2012; Lindemann et al., 2009). However, these additional domains are not clearly recognisable as distinctive separate sets with their own architecture, and they also are comparatively less represented in the literature. Therefore, I have focused in this thesis on the three main domains, especially the process architecture and its intersection with the organisation domain (see figure 1-1).

2.2.5 Relationships between architecture and performance

Previous studies have provided evidence for concrete relationships between characteristics of the architecture within a domain and the project or organisational performance. Table 2-1 provides examples of relationships between the domain architectures and performance measures, making a distinction between composition and structure. The majority of the academic research that has tested and identified empirical relationships between architecture and performance has focused on network structure. In fact, relationships between network composition and performance were found only in the organisation domain, especially in management studies. In addition, most research has considered structure and composition separately, even though a simultaneous consideration provides a more complete description of the system and yields more robust conclusions (Phelps, 2010; Wasserman and Faust, 1989).

Table 2-2: Previously reported effects of network architecture on different domains

	Network Architecture	
Domain and Levels per Domain	**Structure**	**Composition**
Process	-Measures of activity **centrality** have been used to identify **activities constraining** the product development **execution** (Collins et al., 2009). -**Relationships** and trade-offs were found between the **whole process structure** and dependent variables such as **risk, cost, and project duration** (Browning and Eppinger, 2002).	*No study was found establishing a relationship between the network composition of the process domain and performance.* *This thesis provides the means to test for relationships between the network composition of the process and process performance*
Organisation	-Inverted U-shaped effect between network **size** and job **performance** (Chen and Gable, 2013; Tsai, 2001) -High social network **density** increases information flow **efficiency** but reduces **diversity of ideas** (Burt, 1992; Easley and Kleinberg, 2010; Pullen et al., 2012). -High network **density** among a firm's alliance partners **strengthens** the **influence of technological diversity** (Phelps, 2010).	-Positive relationship between **diversity of departments/areas/functions** in the network and **innovation performance** (Jansen et al., 2006; Rodan and Galunic, 2004; Tsai, 2001) -Positive relationship between **diversity of brokerage roles** (Gould & Fernandez. 1989) and organisational **performance** (Gemünden et al., 2007; Tushman et al., 1980) -High **heterogeneity of functional groups** should provide a more diverse pool of knowledge, which could facilitate the development of more **innovative** and systemic design solutions (Jansen et al., 2006; Rodan and Galunic, 2004; Tsai, 2001). -**High heterogeneity** can generate **communicational challenges** derived from the dissimilar knowledge base of the participants and their different perceptions of the design problem, leading to challenges in managing such interfaces (Kleinsmann et al., 2007; Tushman et al., 1980). -High **technological diversity** of a firm's alliance partners increases its **exploratory innovation** (Phelps, 2010).
Product	-Extensive conceptual and analytical exploration about the effects of the **product architecture** on the firm, particularly about the effects of product architecture on the company's **processes and organisation** (e.g. Yassine & Wissmann 2007; Sosa et al. 2004) -Relationship between the presence and **fraction of hubs** and **system's quality:** Evidence that the presence of hubs in a system's architecture is associated with a low number of defects. Also complex engineered systems may have an optimal fraction of hub components. (Sosa et al., 2011).	*No study was found establishing a relationship between the network composition of the product domain and performance*

2.3 The architecture of the process domain

All work is a process. If you want better results at the end–the output–then focus on the process that delivered the results. Any process can [be] managed to be more effective. (Cooper, 1994)

Because of its non-material and transient nature, the process domain can be represented and understood only through models (Smith and Morrow, 1999). Such models examine the design process either 'as is', with a descriptive emphasis on what is 'actual', or they examine the process 'to be', with a prescriptive emphasis on planning (Browning et al., 2006). However, the distinction between the actual and the planned process, in addition to the process's social and technical dimensions, is not always feasible to determine because of the methods and data employed to build models. For example, in standard activity-based process models, typically described either as a flowchart (representing activities in boxes and relationships as arrows) or as a matrix such as a process DSM, certain questions must be considered: What is being represented? Is the model about the planned or the actual process? Does the process architecture illustrate social information flows or technical information dependencies? Unfortunately, the answers to these questions depend on the interpretations of the people who provide information to the modellers, or may not be available at all. Moreover, when providers interpret information requests differently, averaging their answers does not help. As a result, the architecture of process models tends to include a mix of actual, planned, technical, and social dimensions, which renders them more difficult to analyse and interpret.

Looking at the design process architecture from a technical perspective, the process domain is the sphere of knowledge concerned with a set of design tasks and the parameter-driven information dependencies between tasks in the form of information inputs and outputs. In contrast, the social perspective views the process domain as the sphere of knowledge concerned with a set of design activities and the information flows between activities that occur as information exchanges among people (Bucciarelli, 1988; Eppinger and Browning, 2012, p. 130). Following this logic, to identify the information dependencies between tasks, we must draw information from the product domain, in particular the component and subsystem interdependencies. Similarly, to identify the information flows between activities, we must draw data from the organisation domain, particularly information exchanges among people and their participation in design activities. Ideally such information from the product or the organisation domains would be acquired directly, without top-down estimates that rely on information from a few subjects, even if they are experts. However, network-based approaches to the design process that are built only on direct connections between activities or tasks do not explicitly incorporate information about the product or organisation architectures. Instead, these approaches account for the network of people and parameters only implicitly, embedding an estimation of the total effect of the product and/or organisation domain into each of the edges that represents either information flows or information dependencies. Such estimation has the problem of mixing the actual, planned, social, and technical aspects.

In the following sections, I focus on these types of network-based approaches to the design process domains, including approaches at the intersections of process and product architectures and process and organisation architectures.

Features of the design process and their impact on process architecture

Although processes in non-design contexts, such as operations, have been modelled utilising the same network-based approaches used for design processes, (see Alderson [2008] for a review), there are important differences between them. Through a literature review and expert surveys (Maier and Störrle, 2011), we can identify a set of engineering design process characteristics that distinguishing these processes from other human processes.

When design is used to develop solutions to a unique set of constraints and boundary conditions, three features are key distinctive elements. Engineering design processes are **ill-defined**, **iterative,** and **complex**. They are **ill-defined** because their own nature is to solve a problem or challenge without detailed instructions about how to reach the solution or what the solution should look like. They are **iterative** because in the exploration of an ill-defined space, multiple 'alternative paths and successive versions have to be pursued, elaborated, compared, fused, split, improved, evaluated, rejected, and reconsidered …' (Maier and Störrle, 2011, p. 3). As described in section 2.2, design processes are **complex** because of the intertwined relationships of social and technical complexities. Although some processes in operations and manufacturing also are socio-technically complex, those processes tend to minimise or contain emergent behaviours, which are perceived as a negative disturbances. In contrast, the design process–especially when seeking creative, new solutions–requires understanding, harnessing, and taking advantage of inherent socio-technical complexity. In addition, that very complexity, through emergent behaviours, becomes a source of self-organisation and a catalyst of innovation (Alexiou 2010).

As a result of the ill-defined, iterative, and complex nature of the design process, modelling its architecture represents additional challenges compared with other processes. First, the ill-defined nature means that even if an organisation works primarily on engineering design projects related to adaptive or variant designs, each project's design process architecture will change in response to the particular challenges of the design object, the environment, and the natural learning process. Thus, the actual process architecture dynamically revises the planned process architecture in order to follow an ill-defined path. Second, the iterative nature means that unlike standard processes, activities may or not need to be repeated. Moreover, the reason for such a repetition can be either avoidable rework originated on a process problem that shouldn't have happened or a desirable revision needed to improve the design. Static process architecture models that illustrate only a linear sequence of events make these iterations difficult to visualise and analyse. Third, the socio-technical complexity of the design process architecture requires approaches that allow selectively focusing at different levels to contain complexity while maintaining detail.

Fundamental levels of analysis for the process domain architecture

The same three generic levels found in all networks can be applied to analyse process architecture—the node (task or activity), edge (interface as information dependency or information flow), and whole network (overall process) levels (see figure 2-8). Each of these levels is described based on existent approaches to the design process architecture.

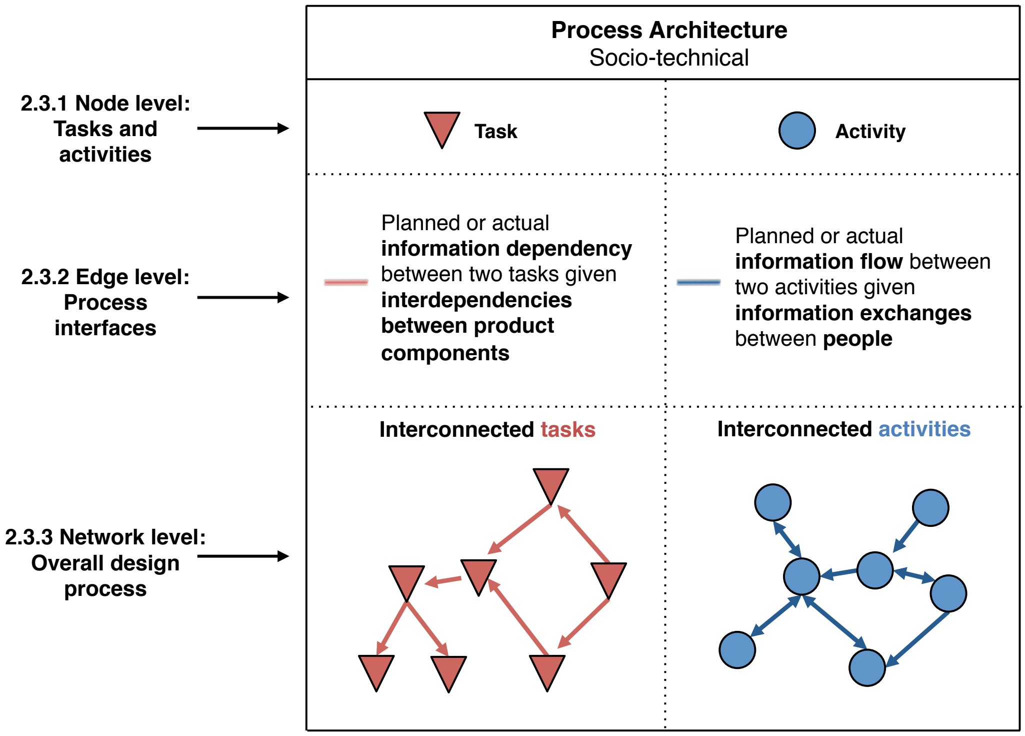


Figure 2-8: Network representation of the process domain

2.3.1 Tasks and activities

Based on the definition of what a process is, detailed process models are built on the basis of tasks or activities. The remainder of this subsection provides operational definitions for distinguishing and classifying the two key elements at this level of analysis: tasks and activities.

The rational for drawing a distinction between design tasks and design activities

As Visser (1992) argued, there is an important conceptual distinction between 'tasks' and 'activities'. Although the two terms are commonly used interchangeably in the process architecture literature, an insufficient conceptual distinction or the lack of a clear working definition can create confusion about the nature of the process architecture being characterised. For example, how can we model and analyse process architecture based on identified, expected, or planned information dependencies? Is this equivalent to modelling and analysing process architecture based on actual information flows? How can we differentiate when we are talking about architectures based on information dependencies, information flows, or a mix of both? Is such distinction relevant? I argue that a conceptual distinction between tasks and activities is instrumental to answer these questions and allows us to advance our understanding of design process architecture through a comparison between the actual and the planned architecture.

Design tasks

A standard definition for a generic task is 'a piece of work to be done or undertaken' (Stevenson, 2010). In design research, authors have offered similar definitions, such as:

> *A task, such as used in empirical design research, is a formulation of a design problem to be solved by [the] participants…* (Bender, 2003).

> *The task concept refers either to what subjects are supposed to do (i.e. their 'prescribed' task, as it has been specified by their manager, by instructions or by manuals), or to the task they set themselves…* (Visser, 1992).

> *…The design problem is decomposed into tasks and sub-tasks. Each task/sub-task with definite goal(s) and time constraints is assigned to appropriate design agent(s). Hence, a design task represents a design effort that must be performed in order to achieve key milestones in a design process* (Sim and Duffy, 2003).

A design task is focused on technical requirements or needs and should specify a concrete and identifiable output. Such technical requirements might be stated verbally, through written instructions or manuals. Examples of design tasks taken from the exploratory case study include: ‘detailing the density of the membrane substrate’, ‘evaluation of the new coating formulation’, and ‘coordination of prototype production between R&D and Manufacturing’. Task granularity can range from a micro-task such as ‘gathering the results from coating sample 12’ to all-encompassing macro-tasks such as ‘R&D of a new flat-sheet membrane’. Each task is explicitly or implicitly linked to a set of technical parameters that must be defined, evaluated, and/or managed (Clarkson and Hamilton, 2000; Wynn et al., 2006).

Therefore, this thesis uses the following working definition for design tasks:

A design task is the work that is required or specified in order to achieve a particular design objective. The objective of a design task can relate to the definition or evaluation of a parameter in the design object, or to the management of the design process.

Design activities

A standard definition for a generic activity is ‘a thing that a person or group does or has done’ (Stevenson, 2010). In the context of design research, definitions are more precise and vary according to setting, but implicitly or explicitly share the same idea: an activity is something a person or group does or has done in response to some sort of task, need, or objective. For instance, definitions that have been previously offered include:

‘(Cognitive) [Design] activity’ refers to the way that subjects actually realize their task on a cognitive level, i.e. the knowledge and other information sources that they use, the way that they make use of them (and of other tools) and other reasoning processes, and their intermediary and final productions (Visser, 1992).

[Design] Activities: The elements of action comprising a process, which in various contexts may be tasks to execute, information to generate, decisions to make, or design parameters to determine. Each activity transforms one or more inputs into one or more outputs. Complex processes are generally broken into phases, stages, or subprocesses, which are further decomposed into activities (Eppinger and Browning, 2012, p. 130).

A design activity is a rational action taken by a design agent to achieve a knowledge change of the design and/or its associated process (i.e. sequence of actions) in order to achieve some design goal (Sim and Duffy, 2003).

> *A design activity is defined as a subdivision of the design process that relates to the individual's problem solving process. It is a much finer division than a stage, covering a shorter period of time. A typical characteristic of an activity is that it reoccurs several times in a process. To categorize the characteristics related to activities, the following activities have been distinguished: generating; evaluating and selecting; modifying; documenting; collecting information; using methods and tools* (Blessing, 1994, p. 10).

Therefore, activities respond to a task description (and in turn, its technical parameters and the design object), but also are likely to be affected by a number of other factors such as changes in the environment, exogenous constraints, characteristics of the individual(s) performing the activity, interactions between individuals, influences of other activities, and so on. In the broad sense, this description is consistent with the premises of Activity Theory, which stresses the importance of activity 'situatedness', including social interactions, division of labour, the specific purpose (task), and the influences of the object towards which the actions are directed (Blackler, 1993; Engeström et al., 1992). However, in this thesis I do not follow Activity Theory's precise definition of an activity, which is restricted to something associated with only one individual, nor do I follow Activity Theory in terms of the relationship and distinction it establishes between tasks and activities (e.g. Bedny & Karwowski 2004). The main reason for this departure is to be consistent with previous design process research, particularly process architecture models, in which activities become a group level construct as analysis becomes more aggregated.

Therefore, this thesis uses the following working definition:

A design activity is a construct that refers to the actual realisation of a particular design task. It involves actions executed individually or in a team to transform a set of information inputs into a set of information outputs.

This short definition allows us to link and draw a distinction between activities and tasks, is compatible with the previously introduced definitions, and highlights the following features:

- Activities can involve an individual or an entire group.
- Activities and tasks are mapped in a one-to-one relationship (when analysed at the same level of detail).
- Activities transform information, and therefore, have a number of inputs and outputs (as information exchanges). Following the definitions provided in section 2.1.1, such inputs and outputs can be summarised as information flows between activities

The distinction between task- and activity-based process models

Although an activity and a task might have the same label, they actually represent different concepts. Consequently, task-based and activity-based process models require different approaches for construction and analysis.

If we use the previous definitions, a process architecture based on a task network is a representation of information dependencies between tasks with interdependencies in technical parameters or needs. In turn, a process architecture based on an activity network characterises information flows between activities with information exchanges between project participants.

Decomposition and aggregation of design tasks and design activities

With this background, I advance to a more detailed classification of design tasks and activities, based on their level of detail (also called granularity) and the nature of the associated design work. Although the classification presented below describes activities, the same structure can be used to classify tasks because each activity can be identified with a corresponding task that defines a specific activity's requirements.

Three broad activity categories can be identified (Parraguez et al., 2014) for the functions activities perform, which builds on the approach by Sosa et al. (2003) to identify and name modular and integrative subsystems. The first category of engineering design work activities has to do with specific modules or subsystems under development, called *modular subsystem activities*. The second category of activities, called *integrative subsystem activities,* are those related to modules or subsystems that have the objective of integrating two or more modular subsystems. A third category corresponds to activities that support, manage, and coordinate design work, called *integrative work activities.* This category is not included in Sosa's (2003) product architecture work, but was considered important in Sim & Duffy (2003) and defined in Parraguez et al. (2014).

In addition to these three broad categories, activities in large engineering design projects can be grouped based on cohesive work packages associated with each subsystem being designed. For example, an automobile's chassis, powertrain, and climate control form distinctive work packages that are designed semi-independently and then integrated at one or more points during the design process (Sharman and Yassine, 2004). The climate control design could be one activity group categorised as modular subsystem activities. The design of the powertrain and the chassis would each be an activity group, categorised as integrative subsystem activities (because their design process is likely to interact with that of multiple other subsystems). In turn, design activities such as project management or test and integration would each be an activity group categorised as integrative work activities. Figure 2-9 shows a generic process domain structure based on the three categories discussed and the overall process breakdown.

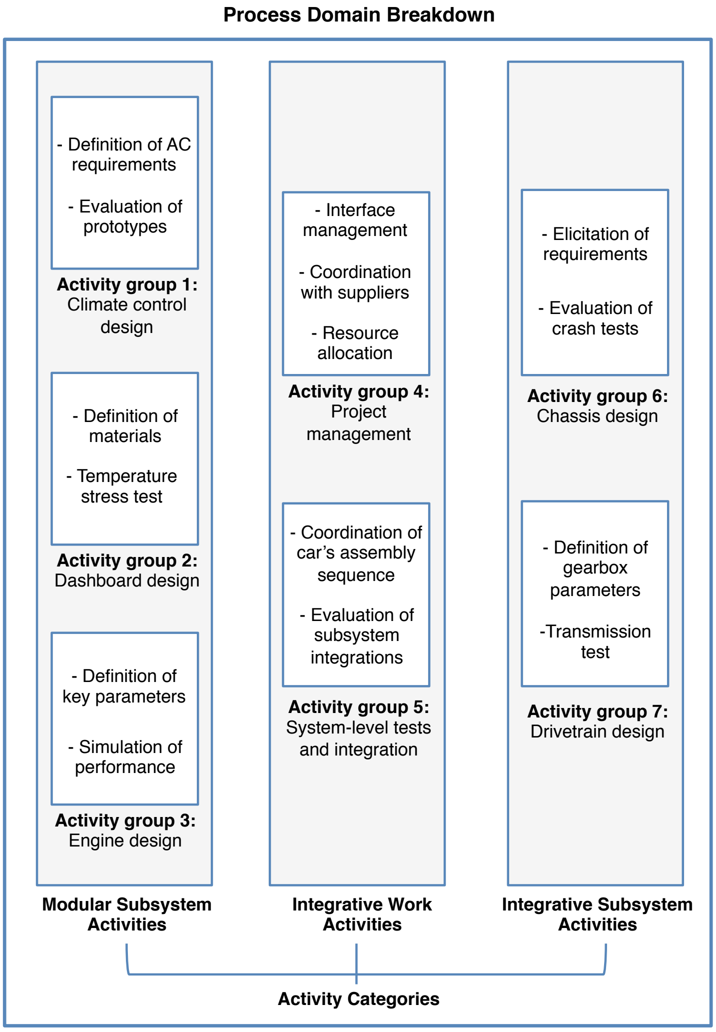


Figure 2-9: Conceptual breakdown of activities using activities related to the design of fictional car

Following Sim and Duffy's (2003) taxonomy, three types of activities can be distinguished in addition to the categories and groups represented in figure 2-9. This additional typology applies directly to each design activity inside an activity group and is based on the type of design work associated to the activity. These activity types are: design definition activities, design evaluation activities, and design management activities. Sim and Duffy (2003) describe them as follows:

Design definition activities ‘seek to manage the complexity of the evolving design while increasingly defining it, until it has all the details required for production’.

Design evaluation activities ‘seek to analyse and evaluate the feasibility of potential design solutions and, by discarding infeasible solutions, reduce the design solution space’.

Design management activities ‘seek to manage the complexity of co-ordinating activities related to an evolving design and its process’.

2.3.2 Process interfaces: information dependencies and information flows

The same distinction introduced for tasks and activities is also relevant when describing and analysing network-based models in the process domain at the ‘edge level’. An edge between tasks represents the actual or planned, parameter-driven information dependencies between those tasks. An edge between activities represents the actual or planned information flows between those activities (Eppinger and Browning, 2012, p. 130). Network-based process models often define the edge between tasks or activities as a single, non-decomposable entity (Clarkson and Hamilton, 2000), sometimes associating attributes to this edge such as intensity or frequency. However, such a description is insufficient to allow for a detailed characterisation and analysis of each edge’s structure and composition, or to think about actual interfaces instead of plain edges between tasks or activities.

Interfaces

Integration of design efforts occurs at process, organisational, and product interfaces (Clarkson and Eckert, 2005, p. viii; Eppinger and Browning, 2012; Rechtin, 1990). Problems at these interfaces often result in failures in the designed engineering system and add uncertainty to the design process (Felekoglu et al., 2013; Maier, Eckert, et al., 2009). Consequently, an understanding and active management of interfaces is essential for design process improvements, particularly with the design of large engineering systems in which design activities and tasks cover different subsystems that include hundreds or thousands of design engineers (Browning, 2009; Madni and Sievers, 2014; de Weck et al., 2011).

Defining interfaces and interface management

Although the term *interface* has different meanings depending on the context—and a precise operational definition for each context often is not provided—a set of characteristics can be used to define the term across multiple contexts including products, processes, and organisations. These characteristics are summarised as:

- An interface connects, or allows for the connection of, two distinctive elements or groups (Morris, 1997; Stevenson, 2010).
- The elements on each side of the interface have or require some sort of interaction (Loch and Kavadias, 2008; Morris, 1997).
- An interface between two elements or groups can consist of only one element, a set of them, or a complex system in its own right (Buede, 2009, p. 61).
- An interface does not need to be material, but may be information, a concept, or a combination of material and immaterial components (Lakemond et al., 2007; Morris, 1997).
- An interface might be permanent or temporal, and its existence depends on the presence of two interacting elements or groups at each side of the interface (Morris, 1997).

Based on these characteristics, a dependency or required interaction between two components, departments, or activities in the form of an edge does not constitute an actual interface, although identifying these edges may be necessary for planning purposes and to narrow data collection and analysis.

If we concentrate on activity models, an interface enables the information flow between activities, effectively connecting a pair of activities and fulfilling their information dependencies. Therefore, an interface between two activities may consist of people, information technology platforms, other resources facilitating the information flows between activities, or any combination of these, which is consistent with several studies (Christian, 1995; Durugbo et al., 2011; Morelli et al., 1995; Sosa et al., 2007b).

Interfaces between activities traditionally have been hard to characterise for at least two reasons: 1) The elements on each side of the interface are transient, even more so than people, because activities typically have a beginning and an end during the project's lifetime, and 2) an activity is not a tangible element, but rather a notion that combines tangible elements, such as people and design parameters, found outside the traditional boundaries of the process domain.

Interface management and interface problems

Interface management can be broadly defined as the management across a common boundary (the interface) of interactions that happen between and/or within interdependent elements of the organisational, product, or process domains. In engineering design, the most common interface management occur in the product (e.g. Rahmani & Thomson 2011; Maurer 2007; Bruun et al. 2013) and the organisation domains (e.g. Maier, Kreimeyer, et al. 2009; Sosa et al. 2007b; Eckert 2001). However, process domain interfaces have not received the same level of attention.

According to some research studies (Browning et al. 2006; Browning & Ramasesh 2009; Browning 2009), an issue that affects our understanding of interface management is that process models emphasise design tasks and their information dependencies rather than how information is delivered and transformed between each pair of activities. In addition, most process models consider only planned or expected information dependencies between tasks, not actual information flows or actual work performed at the interface between two activities. This limitation of process interfaces likely stems from their challenging nature and the level of analysis that current process models typically apply, in which attention is on the whole activity network, not individual interfaces. Clearly, new approaches are needed to provide appropriate support for interface managers of complex engineering design projects and to better understand the potential sources of interface problems.

As previously defined, an interface connects or allows for the connection of two distinct elements or groups (Morris, 1997). Consequently, **interface problems** occur when the performance of the connection between activities (the interface) is lower than expected, which hinders the interaction between two elements or groups. More specifically in the process domain, interface problems equate to interaction issues between interdependent activities that hinder the performance of at least one of the involved activities, and therefore, its outputs (Eppinger and Browning, 2012; Heisig et al., 2010).

2.3.3 Overall design process

The combination of the two previously introduced levels (activities and interfaces) allows a whole network of design tasks or activities to be assembled based on the information dependencies or information flows between them. From an academic and a managerial perspective, an essential tool for complex engineering design projects is the ability to quantify, analyse, and understand the evolving information flows between activities (Eckert et al., 2005; Eppinger and Browning, 2012). The intended or expected evolution of information flows between activities (based on tasks' information dependencies) has been modelled and analysed through task-network approaches, such as the design structure matrix (DSM) (Eppinger et al., 1994), workflow diagrams, IDEF, CPM/PERT, or Petri nets. In turn, the evolution of the whole process often is framed and guided through some variant of stage-based models (Gericke and Blessing, 2012; Wynn, 2007). However, to quantify and analyse how information actually flows between activities, we require a model that simultaneously integrates the dynamics of process and organisation architectures.

Before Parraguez et al. (2014), studies of the design process had not provided or empirically tested a model that could analyse the evolution of information flow between activities in a way that clearly distinguished actual flows from information dependencies or intended information flows. As a consequence, it had not been possible to compare actual information flows against expected or idealised information flows at each project stage and point of time. This gap was not only a shortcoming in overall knowledge about the design process, but also a hindrance to monitoring projects' overall progress and to active benchmarking.

Temporal dynamic of the overall design process

In terms of temporal evolution, the process domain has been mostly described and analysed by sequencing design tasks and analysing temporality at the level of design stages (Blessing, 1994; Wynn, 2007, p. 17).

At the level of design tasks, process dynamics have been materialised in the description of a sequence of tasks from which a process temporality can be deduced, analysed, and optimised, if necessary (e.g. Meier et al. 2007; Campos Silva et al. 2012; Eppinger et al. 1994). This view allows for a comprehensive computational analysis of the time dimension; however, it often does not represent the temporality of the actual design process, especially in process DSM approaches. Instead, it shows a chain of dependencies that can be used to plan an appropriate task sequence, and therefore, organise the design process and activities more rationally.

At the level of design stages, process dynamics tend to be associated with generic and prescriptive models of the new product development process (e.g. Ulrich & Eppinger 2012; Hubka 1982; Pahl et al. 2007) and to systems engineering models depicting the design process's logical progress (e.g. Haskins et al. 2011). Unlike specific activities or tasks, these stages are easier to generalise because they are less detailed. As a consequence, they can be used as general benchmarks or guides with which to compare or measure a project's progress.

Design process stages

Staged-based models of the design process reflect the dynamic nature of transforming a set of requirements into a detailed instructions for building or implementing the design object (Simon, 1996; Ulrich and Eppinger, 2012). As the design process unfolds in stages, information flows between activities evolve. This evolution can be traced to temporal and co-dependent aspects, such as the progression of the design object (Ulrich and Eppinger, 2012) and the maturity of the design process (Maier et al., 2008).

To facilitate discussion, I focus on the overall stages described in Ulrich and Eppinger (2012) and apply the system-development perspective found in INCOSE's systems engineering SE-V model (Haskins et al., 2011). I selected these models because they provide widely accepted, generic descriptions of new product development and system engineering processes. In addition, and as described in Howard et al. (2008), there are multiple commonalities between the stages in these models and those in other well-known engineering design process models.

Figure 2-10 offers an overview that works as a reference point for the characterisation of each stage. The focus of this thesis lies between the stages of conceptual design and system integration, which are the limits of the predominant focus of engineering design (Clarkson and Eckert, 2005, p. 5). Consequently, aspects like strategic planning and implementation are not explicitly covered here; however, if necessary, those two stages could be included inside conceptual design and system integration respectively, because they are in relative terms at the same levels of integration/decomposition.

Combining the descriptions for the SE-V model and Ulrich and Eppinger (2012) stages, each stage can be summarised by level of decomposition or integration, level of abstraction or maturity of the design object, and the key activity categories expected to dominate the stage:

Conceptual design: Individuals from multiple functions contribute inputs in the context of tasks, such as idea generation, selection of concepts, and the preliminary planning of technical specifications (Ulrich & Eppinger 2012). At this stage, especially if the engineering system to be developed is relatively new for the company, problems and activities will be ill-defined, and if the stage is poorly managed, the required convergence to guide the work and the subsequent stages may not be reached (Austin et al., 2001). For systems engineering, this stage is defined by low decomposition, high abstraction, and is typically dominated by integrative work activities.

System level design: The overall architecture agreed upon in the conceptual design stage is defined in detail, including the decomposition of subsystems and components (Ulrich & Eppinger 2012, p.15). Preliminary engineering begins with a division of the work into multidisciplinary teams, assigned first to a core of relatively integrated subsystems that require high levels of coordination (Ulrich, 2011, p. 88). In addition to major subsystems and interfaces, this stage comprises what Pahl and Beitz (1996) called 'embodiment design', which includes the first technical drawings of the overall system architecture. In terms of systems engineering, this stage is defined by a low- to medium-level of decomposition, a medium level of abstraction, and a combination of integrative work activities and modular subsystem design activities.

Detailed design: The complete set of specifications for all components is defined at the highest level of decomposition and detail (Ulrich & Eppinger 2012, p.15). Results of this stage often include standard parts to be acquired from suppliers and the first inputs for fabrication. As the degree of technical specialisation reaches its peak, the subsystem teams work more independently and in a relatively modular fashion. This stage reaches the highest level of decomposition, lowest level of abstraction, and is typically dominated by modular subsystem design activities and integrative subsystem design activities.

System integration: All modular subsystem activities must integrate their results, and components must be tested and validated at the system level, which is why this stage is sometimes called 'testing and refinement' (Ulrich and Eppinger, 2012; Ulrich, 2011). If integration problems are detected, one or more components from any subsystem may need to be reworked, which can create a cascade of iterations. Depending on the issues identified (or opportunities for improvements), this stage can be relatively simple and quickly move to implementation, or require complex, time-consuming iterations. In systems engineering terms, this stage has the highest levels of integration and design maturity, and is typically dominated by integrative work design activities and integrative subsystem design activities.

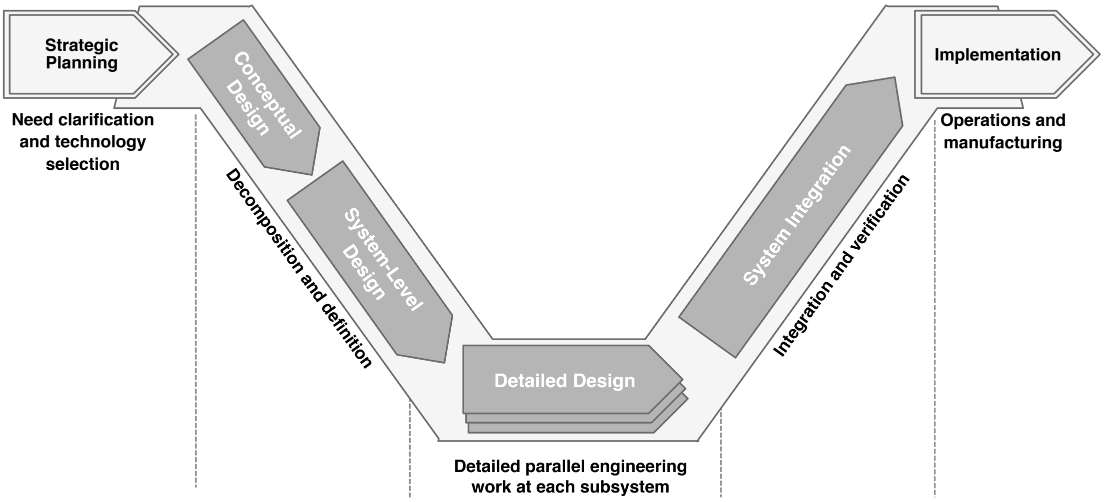


Figure 2-10: System engineering V model used in this thesis. Adapted from Ulrich and Eppinger (2012) and INCOSE's SE-V model (Haskins et al. 2011)

2.3.4 Process architecture: features covered by design process models

Previous literature reviews on the design process point out numerous existing design process models, which have a number of commonalities (e.g. Blessing 1994; Wynn 2007; Maier & Störrle 2011; Gericke & Blessing 2012; Browning 2009). Based on common architectural aspects of activity- and task-based models, table 2-3 summarises how these approaches cover the three levels of analysis as well as structure and composition. (These approaches are not explicitly at the intersection of other domain architectures, such as product or organisation).

Table 2-3: Summary for the three levels of analysis and the architectural aspects of structure and composition covered by existing activity and task based models

	Node: Tasks and activities	**Edge:** Information dependencies and flows	**Whole Network:** Task and activity networks
Structure	**Tasks** and **activities** are represented as **non-decomposable nodes**. As a result, it is not possible to characterise their internal architecture or structure.	**Information dependencies** and **information flows** are represented as a **single edge**. As a result, it is not possible to characterise their internal architecture or structure.	Network analysis has been used to quantify **centrality and other network metrics of individual nodes** embedded in the whole network. In addition, the same type of network analysis has been used to **describe the architecture of the whole network** (e.g. Collins et al. 2009; Braha 2006; Kreimeyer & Lindemann 2011). The **dynamic analysis** of task networks can be made, based on reported or calculated sequences between tasks.
Composition	Models provide only a **compositional characterisation for the task or the activity as a whole**. Such characterisation may be an assigned attribute or calculated based on the whole network structure. Attributes include cost, risk, duration, and criticality (e.g. Browning & Eppinger 2002).	The reviewed models provide only a **compositional characterisation for the information dependency or flow as a whole.** Attributes are assigned to quantify the edge, based on the reported strength, probability, frequency, perceived importance, or criticality (Eppinger and Browning, 2012, p. 139).	The composition of network-based process models has been expressed in terms of the **type of tasks or activities.**

Summary

The design process, analysed from the perspective of the process domain and its architecture, has been a fertile research area in which a number of descriptive and prescriptive models and methods have been developed. Most of these efforts can be characterised as an activity or a task network, and therefore, can be studied in terms of structure and composition. Despite the number of existing design process models, and as anticipated in subsection 2.1.4 (models and the design process), analysis and interpretation are hindered by the lack of detailed descriptive models of the actual design process that can create a clear distinction between what is planned and what is actual, what is social and what is technical.

This thesis argues that a reason for this gap is because most models do not fully address the design process as a socio-technical system of information transformation. To do so would require modelling each activity as a task performed by one or more people, and the process as a collection of interdependent activities connected by people who collectively exchange and transform information. Based on this argument, a detailed and descriptive model of the design process must take into account the dynamic organisation network that implements the process and a mapping of people to activities. In section 2.5, I review models that explicitly integrate elements of the product and/or the organisation architectures into the process domain. The goal is to identify elements that enrich the description and analysis of process architecture and explicitly address the socio-technical nature of the design process.

2.4 The architecture of the product and the organisation domains

Although the product and organisation domains on their own are outside this study's scope, the intersections of these domains' architectures with the process domain are useful for understanding process models. This section introduces key elements of the architectures of the organisational and product domains and their levels of analysis, which can be used later in the context of the intersection of process and organisation domains as well as the intersection of process and product domains.

2.4.1 The product domain

The product domain represents the sphere of knowledge concerned with the actual or planned interactions between components (and subsystems) of the engineering system being designed. Such interactions, which can be material, spatial, energy flows, information flows, or so on (Eppinger and Browning, 2012, p. 18), determine the behaviour and function of what is designed.

The network architecture of this domain, represented in figure 2-11, consists of components (node level), planned or actual interactions between components (edge level), and the network of interconnected components that constitute the engineering system (whole network level).

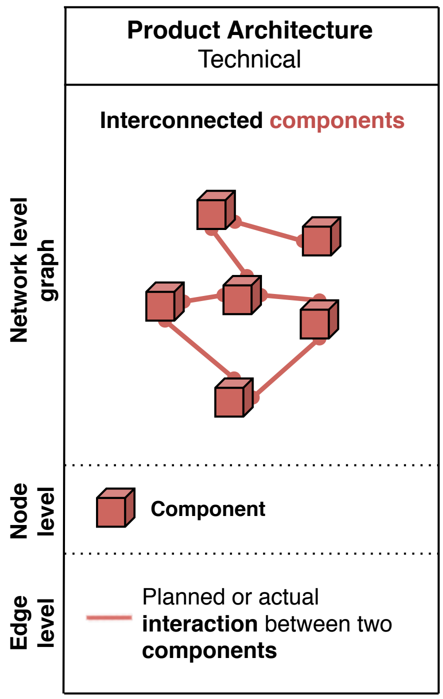


Figure 2-11: Architecture of the product domain

At the structural level, the architecture of the product domain can be characterised by the interconnectedness between components and/or subsystems of the engineering system (depending on the level of detail of the analysis). For example, Sharman and Yassine (2004) proposed a systematic approach to characterise complex product architectures, based on their structural characteristics. Key elements included the type of interaction between the components, the relative weight or intensity of the interaction, and any directionality in the interaction.

At the compositional level, the type of components and the interactions between components generate a distinctive combination characterised according to compositional diversity, which alongside structural measures, can indicate the product complexity (e.g. Wyatt et al. 2012; Sharman & Yassine 2004).

In network terms, the product domain is typically classified and analysed as a static network (Browning, 2001), but we can map the build-up of the product architecture based on the sequence of interactions or logical assembly of product components, which can be used, for example, to support design for assembly approaches (e.g. Moultrie & Maier 2014).

2.4.2 The organisation domain

In the design of engineering systems, the organisation domain is the sphere of knowledge concerned with the actual or planned information exchanges between people involved with the design process. By allowing the flow of information between interconnected activities, these interactions ultimately make the design of an engineering system possible (Allen and Henn, 2006; Eppinger and Browning, 2012, p. 80).

The network architecture of this domain, represented in figure 2-12, consists of people (node level), planned or actual information exchanges between people (edge level), and the network of interconnected people (the whole network level).

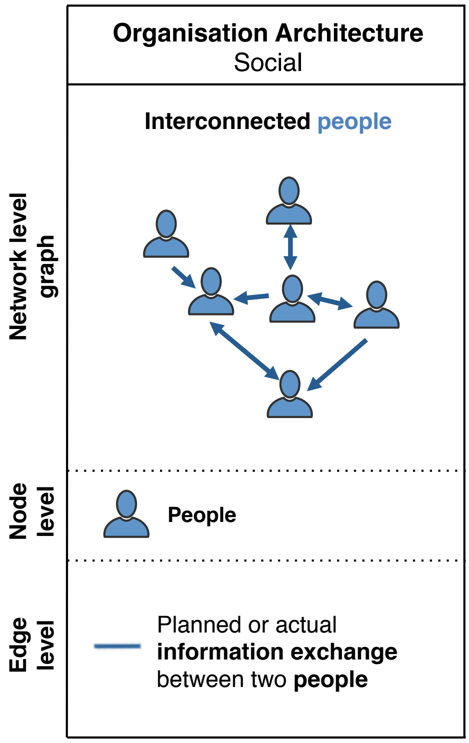


Figure 2-12: Architecture of the organisation domain

At the structural level, the architecture of the organisation domain characterises the interconnectedness between people or groups of people. The complexity of this interconnectedness can be broken down and described. Key elements of this breakdown are the 'type of relationship' between people, the relative 'weight' of the connection or relationship, and the existence of 'directionality' in the relationship.

At the compositional level, network approaches describing organisation architecture include key people (or groups such as departments), therefore the organisation domain's compositional characteristics may be described according to the 'attributes' of individuals or groups in the organisation. For example, any group or whole organisation will include a combination of people from different departments, with various functions, backgrounds, and seniority levels, and these combinations can be measured by their compositional diversity.

Organisation domain models, based on relationships such as communication, information exchanges, and advice networks, have a natural temporality, whether accounted for or not. In contrast, formal roles and other top-down descriptions of the organisation may rarely change or not change at all during the relevant study period. However, regardless of the intrinsic temporality of the model's content, most network-based organisational models used in engineering design and R&D have been only static. This static nature can be explained by a combination of reasons related to conceptual models, methods, available data sources, and computational difficulties. In fact, a commonly accepted classification of DSM models categorises by default all organisation architecture as static models (Browning, 2001; Eppinger and Browning, 2012, p. 11), perhaps because the dynamic evolution of the structure is difficult to reflect and analyse appropriately in matrix-based models. One way to capture this evolution is to compare matrices that reflect different time periods (Eppinger and Browning, 2012, p. 99) and then use a method such as delta DSMs (de Weck, 2007) to compute and analyse the differences. However, this approach is limited in terms of applicable metrics and tend to be impractical when applied to dozens or hundreds of time frames.

In contrast to matrix-based models, graph-based models are more flexible to incorporate the time dimension, and therefore, may be more suitable for a dynamic analysis of the organisation domain. However, only recent advances on network analysis models, methods, and software have allowed more widespread use of this technique. As a result, most dynamic analyses of the organisation domain lie outside the specific field of engineering design and new product development. Exceptions are found in recently published works such as Jepsen (2013) and Cash et al. (2014).

2.5 The design process at the intersection of product and organisation architectures

Previous studies have explored the relationship and alignment between the architectures of the product, organisational, and process domains, testing what is sometimes described as the 'mirroring hypothesis', which suggests a desirable and natural structural correspondence between the architectures of the three domains (e.g. Colfer & Baldwin 2010). For example, Sosa (2008) showed that such alignment occurs in practice through a mix of intra- and cross-domain interfaces, which can be operationalised through the combination of a communication and a process-organisation affiliation matrix (Sosa, 2008). This line of work has addressed such questions as 'Who *should* talk to whom?', 'Which interfaces *should* they talk about?' (Sosa, 2008), and has helped to assess whether the degree to which various interfaces are attended, unattended, or unanticipated (e.g. Vignoli et al. 2013; Sosa et al. 2004).

Instead of examining the alignment between architectures of different domains, other studies have integrated architectural information outside the process domain in order to derive an enriched or more accurate description of the process architecture. For example, the Signposting Framework (Clarkson and Hamilton, 2000; Wynn et al., 2006), and other approaches at the intersection of process and product architectures, allow to identify the sources of information dependencies between tasks (e.g. Senescu et al. 2012). This identification provides an enhanced view of the process architecture based on more detailed information dependencies. Along the same lines, Gokpinar et al. (2010) found substantial evidence for the relationship between the centrality of components in a product architecture and the quality of components (inverted-U relationship), and also a significant effect on quality of mismatches between product and organisation architecture.

In both types of studies, the key is that potentially increased understanding of the design process lies at the intersection of architectures. In the following three subsections, I explore in additional detail studies that have examined the intersection of product and process architectures (2.5.1), process and organisation architectures (2.5.2), and other generic approaches to model and analyse cross-domain architectures (2.5.3). Based on the research objectives in this thesis, certain aspects of these studies are particularly relevant, including: the ability of these approaches to compare actual and planned process architectures, the operationalisation of a distinction between planned and actual process architecture, the main data inputs, the degree to which the architecture at multiple levels can be examined, the incorporation of process dynamics, the characterisation of structure and composition, and other aspects related to the analytical methods, such as the possibility of using valued matrices, capturing indirect paths, and establishing relationships between process architecture and performance.

2.5.1 Design at the intersection of process and product architectures

Design approaches at the intersection of the product and process domains allow us to map information dependencies between tasks with a higher degree of detail because of the explicit inclusion of product parameters. Dynamic task models, particularly the Signposting Framework (Clarkson and Hamilton, 2000; Wynn et al., 2006), 'view design as a dynamic process organised around the changing state of the product' (Wynn, 2007, p. 41). In the Signposting Framework, a detailed description of both the planned and the actual process architectures (from a technical perspective) can be captured through indirect mapping of activities via the network of interdependent parameters. However, this approach does not have a direct equivalent for design activities.

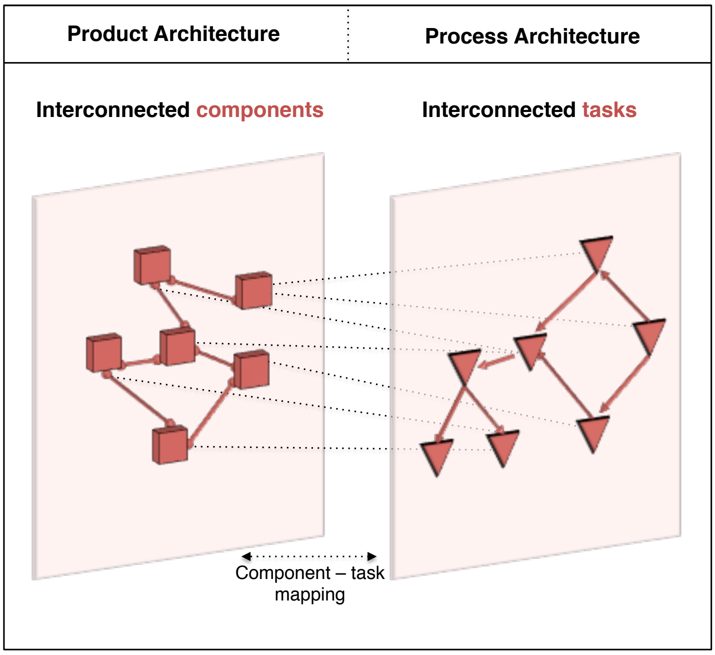


Figure 2-13: Graphical representation of the intersection between the product and process architectures.

Figure 2-13 provides a graphic example of the intersection between these two architectures.

Information dependencies between tasks

Unlike other parameter-based process models that connect tasks directly through a single edge, the Signposting Framework (Clarkson and Hamilton, 2000; Wynn et al., 2006) connects tasks indirectly through a network of interdependent design parameters between them. If two tasks do not possess interdependent parameters, then they are not linked. If they possess at least one such parameter, then a dependency is identified. As the authors of this framework have stated, other network-based process models based on DSMs, PERT, IDEF0, IDEF3 (or other task networks) also can describe the dependency between two tasks given interdependent parameters, but only implicitly. Therefore, the explicit inclusion of the parameters in the Signposting Framework provides additional information about the actual nature of the information dependency between tasks, which increases the model's overall accuracy. Moreover, although mapping tasks in this way might be more time consuming, it does not rely so heavily on the tacit process knowledge of those who describe the dependencies. Also, this systematic way of mapping information dependencies between tasks leverages the increasingly available detailed information about the product architecture and the dependencies between components and their parameters.

Another approach, that follows a similar logic is the Automatic Information Dependency Algorithm (AIDA) (Senescu et al., 2012), which uses interdependencies between project engineering files as a proxy of interdependent technical parameters. With that data and assumptions about the relationship between opened and interdependent files, AIDA can recreate a network of information dependencies between tasks.

Table 2-4 presents a summary of key aspects of the Signposting Framework and AIDA. Although these are not the only two approaches that use the intersection of the architectures of the product and the process domains, they are good examples of this type of study.

Table 2-4: Summary of approaches at the intersection of product and process architectures

Frameworks and approaches	Overall objective	Comparison between planned and actual process architecture?	Operationalisation of an architectural distinction between information dependencies and information flows?	Main data inputs	Seeks to obtain		Features Architecture				Method		
					Information flows, information dependencies or both?	Planned architecture, actual architecture or both?	Multilevel distinction between the individual architectures of activities, interfaces and whole process?	Dynamics	Structure	Composition	Valued direct and indirect paths	Identification of outliers	Architecture - Performance test
Generating a network of information dependencies automatically (Senescu et al. 2012)	To infer a network of information dependencies in real-time by capturing how professionals interact with files	No, but enables the quantification of the actual process architecture through the product architecture	No	Data logs with opened files	Information dependencies through interdependent files	Actual architecture with focus on information dependencies	Whole process only	Yes, for the whole process network	Allows to quantify whole process structure	No	Valued matrix. Is possible to measure direct and indirect paths	In the form of unexpected dependencies	Posed as a possibility
The Signposting Framework (Clarkson & Hamilton 2000; Wynn et al. 2006)	Provide a detailed parameter-based process model and framework to study information dependencies between tasks	No, but allows mapping actual information dependencies and enables comparisons against planned information dependencies	Detailed operationalisation of actual information dependencies but not of information flows	Product DSM with detailed parameter interdependencies and mapping between parameters and tasks	Actual information dependencies through interdependent component parameters	Actual process architecture based on information dependencies	Allows analysing at the whole process, task and interface levels	Yes	Allows for structural analysis at each level	Composition can be studied but is not detailed explicitly in the framework	Can be valued. Is possible to measure direct and indirect paths	Possible based on project simulation and other benchmarks	Posed as a possibility

2.5.2 Design at the intersection of process and organisation architectures

As figure 2-14 illustrates, approaches at the intersection of process and organisation allow information flows between activities to be mapped in more detail because of the explicit inclusion of people. The closer these information flows resemble reality, the better they will describe the actual process architecture obtained through the model. As information flows are intrinsically dynamic, static models lose details regarding the evolving actual process architecture; however, most models at the intersection of process and organisation are classified as static, and therefore share this same limitation.

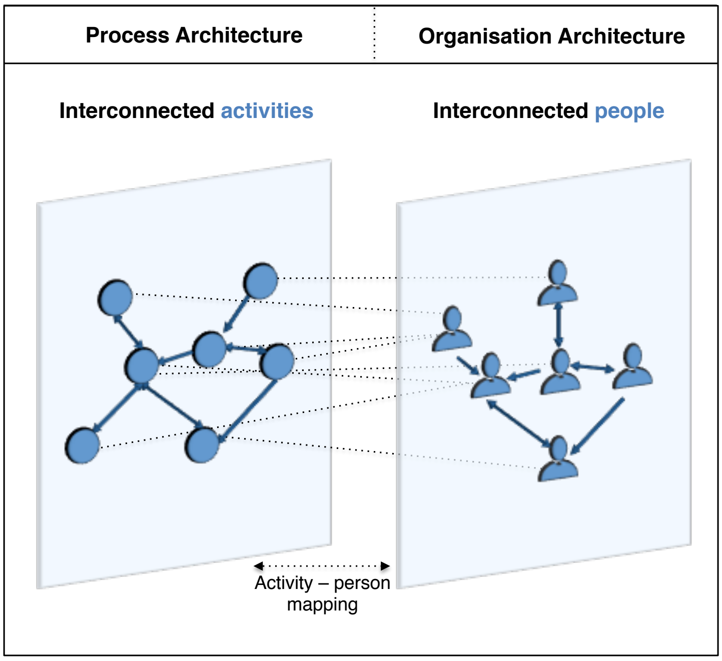


Figure 2-14: Graphical representation of the intersection between the process and organisation architectures.

Static models provide an aggregated view of the information flows between activities through one or a few snapshots. Some of these models use a single matrix and are limited to a one-to-one mapping of activities and people or organisational units, such as 2D DSMs (Morelli et al., 1995). Other studies use cross-domain matrices, including Domain Mapping Matrices (DMMs) (Danilovic and Browning, 2007; Maurer, 2007; Yassine, Whitney, et al., 2003), affiliation matrices (Sosa, 2008), their equivalent Multiple Team Membership matrices (MTMs) (Vignoli et al., 2013), and bimodal network-based approaches (Durugbo et al., 2011) that allow for many-to-many mapping. Unfortunately, the predominantly static way in which these models calculate information flow metrics for each time period makes it difficult to dynamically contrast those measures with prescriptive design process stages or with the result of methods that prescribe an idealised sequence of activities.

Dynamic models were not found that could simultaneously consider the evolution of process and organisation architectures and characterise the resultant network of information flows between activities. Although Christian (1995) developed a simulation-based approach to information flows that does consider dynamics and includes the two domains, that model's emphasis is on computational simulation rather than on capturing and modelling actual information flows between activities. Therefore, the model does not have a direct application for mapping actual process architecture unless all necessary parameters for the simulation are known.

Information flows between activities

An edge between two activities represents the existence of an actual or planned information flow between them. Process models that do not consider information from the organisation domain cannot characterise the edge beyond the limited information embedded in a single line connecting the two activities. Durugbo et al. (2011) offered an alternative to this limitation in an approach to model collaboration using complex networks. In their approach, information flows between activities can be decomposed and examined through the information exchanges between the people involved in the process. However, Durugbo's approach does not offer an explicit characterisation in terms of the network structure and composition of the edges between each pair of design activities, but rather, focuses mainly on measures at the node level for each of the participants and activities (organisation architecture focus). An approach that moved from a representation of a single edge to a richer description of how information flows between activities could yield an activity-activity edge as a social interface between activities, with its own structure and composition.

Table 2-5 presents a summary of key aspects for some of the most relevant approaches at the intersection of process and organisation architectures found in this literature review. With the exception of Christian (1995), all these approaches focus their final analysis on the architecture of the organisation domain rather than an enriched characterisation of the actual process architecture.

Table 2-5: Summary of approaches at the intersection of process and organisation architectures

Frameworks and approaches	Overall objective	Comparison between planned and actual process architecture?	Operationalisation of an architectural distinction between information dependencies and information flows?	Main data inputs	Seeks to obtain: Information flows, information dependencies or both?	Seeks to obtain: Planned architecture, actual architecture or both?	Features Architecture: Multilevel distinction between the individual architectures of activities, interfaces and whole process?	Features Architecture: Dynamic	Features Architecture: Structure	Features Architecture: Composition	Method: Valued direct and indirect paths	Method: Identification of outliers	Method: Architecture - Performance test
Inter-team technical communication (Sosa et al. 2014)	To investigate how informal communication structure (particularly a third party) affects the likelihood of technical communication between interdependent teams	No, but allows to compare informal inter-team communication networks against task interdependence	Indirectly through the delta between "communication flow" and "task interdependence flow"	Product DSM (technical process related interfaces) plus formal and informal organisation DSM	Information flows between functional teams (organisation domain)	Both (task interdependence and organisational communication network)	On the organisation side focus on dyadic and triadic levels	No	Detailed analysis of organisational structure, especially at the triadic level	Through analysis of functional team membership	Binary	Through mismatches between actual communication and technical interdependence	Posed as a possibility through the connection with product performance metrics
Predicting and managing technical interactions (Morelli et al. 1995; Sosa 2008)	Comparing potential interactions with actual communications given the product architecture	No, but allows to compute a matrix of potential interactions and compare it against directly gathered actual communications	Indirectly. Information dependencies can be operationalised through the product/process architecture and information flows through the organisational architecture	Organisation DSM for comparison Org.-Prod. Plus affiliation Matrix and Product DSM (technical process-related interfaces)	Potential information flows in the organisation domain (to be compared against actual)	Both, planned/ potential organisational architecture and actual organisational architecture	Focus on whole process and organisation level architecture plus the possibility of analysing activities and interfaces	No, although it allows for linear sequencing in the matrix	Analysis of whole process and organisational level architecture	Possible though analysis of functional team membership	Yes, although somewhat limited when working with valued matrices	In the form of unattended interfaces	Through analysis of "truly potential unattended interactions"
Building agile design teams (Vignoli et al. 2013)	To improve the creation of cross-functional teams through the identification of critically understaffed component interfaces	No, but allows comparing through an alignment matrix information dependencies between components and organisation architectures	Indirectly. Information dependencies through product/process architecture and information flow through organisational architecture (information exchanges)	Product DSM (technical process-related interfaces), organisation DSM, Org.-Prod. Affiliation Matrix	Both for comparison purposes	Both but only through misalignment	Focus on process and organisation level architectures with possibility of analysing other levels	No	Analysis of whole process and organisational level architecture	Possible though analysis of functional team membership	Yes, although somewhat limited when working with valued matrices	In the form of "critically missing team interactions"	Posed as a possibility
Modelling collaboration using complex networks (Durugbo et al. 2011)	Characterise organisational structures for collaboration and propose indicators to assess organisational behaviour	No, but enables the quantification of the actual process architecture through the organisational architecture	A distinction is made between activity edges (information dependencies) social edges (mapping activity-people) and social edges (information exchanges)	Organisation DSM, Process DSM (information dependencies) and mapping of people and activities	Information flow with focus on the organisation domain	Actual architecture	Whole process only	Network is defined as static/ aggregated	Allows for detailed structural analysis with focus on whole network level	No	Defined for binary only	Through collaboration indicators	Possible but not made explicit
Simulations of information flow between tasks (Christian 1995)	Provide an information flow model between interdependent tasks	No but allows comparisons between the results of the simulation and process plans or other process representations	Yes, information flows and information dependencies are separated	Tasks, dependencies and roles plus detailed process parameters	Information flows and information dependencies (based on simulation)	Only simulated architecture	Structural network analysis at multiple levels is not explicit but could be operationalised	Dynamic simulation	Results can be transferred to a DSM for structural analysis but the focus is on simulation of information flows	No explicit metric for architectural composition but can be incorporated	Valued and indirect paths	Yes, in the form of anomalies or unexpected simulation results	Simulation can make predictions on process performance

2.5.3 Other cross-domain approaches at the intersection of architectures

Other types of approaches do not focus on a particular intersection of architectures, but rather seek to examine and provide new insights at the intersection of different sets of architectures. Because of their flexibility, these approaches are a useful platform to develop models and methods for analysing cross-domain interactions. These approaches can be divided into those that set the conceptual basis and models for cross-domain analysis (e.g. Maurer 2007; Eppinger & Salminen 2001; Yassine, Whitney, et al. 2003), those that seek to quantify the characteristics of these cross-domain architectures (e.g. Kreimeyer & Lindemann 2011), and those generic approaches to process discovery such as 'process mining' (van der Aalst, 2011).

The first type of cross-domain approach lays the foundation for cross-domain analysis in engineering design projects by defining the domains, their architectures, and how their logical interdependencies and cross-effects. For example, Eppinger and Salminen (2001) not only defined the three main domains and elaborated on their dependencies, but also set key hypotheses about cross-domain alignment and evolution that subsequently have been tested and verified.

The second type of approach quantitatively characterises architectures one domain at a time or across domains. A recent good example is the work of Kreimeyer and Lindemann (2011), which provided insights into structural features that affect behaviour through a comprehensive, quantitative characterisation of complex architectures. Taking the models and their generated architectures as a given, this approach aims to quantify architectural complexity, as well as to analyse and interpret the results of those characterisations.

The third approach is more generic and aims to discover actual process architectures and compare them to a given benchmark or performance measure. This approach has been termed 'process mining' (van der Aalst, 2011) because of its data-driven, bottom-up nature. With the increasing availability of process-related big data, process mining has become a reliable alternative to top-down models that rely on experts' judgments. However, despite its potential, process mining has been used mainly for non-design business processes. Although not explicitly associated, approaches like AIDA use process-mining principles, such as the processing event-logs, to discover the otherwise hidden architecture of a process. Table 2-6 presents a summary with these three cross-domain approaches.

Table 2-6: Summary of other cross-domain approaches at the intersection of architectures

Frameworks and approaches	Overall objective	Comparison between planned and actual process architecture?	Operationalisation of an architectural distinction between information dependencies and information flows?	Main data inputs	Seeks to obtain		Features Architecture				Method		
					Information flows, information dependencies or both?	Planned architecture, actual architecture or both?	Multilevel distinction between the individual architectures of activities, interfaces and whole process?	Dynamic	Structure	Composition	Valued direct and indirect paths	Identification of outliers	Architecture - Performance test
Multi domain analysis of engineering design projects (Maurer 2007; Eppinger & Salminen 2001; Yassine et al. 2003)	To model and analyse the structure of process, product and organisation architectures (as well as their intersection) to generate new insights through increased structural awareness	No but enables the quantification of any network-based process model facilitating comparisons	No, but sets the basic elements to intersect domains	Any data on interactions in the context of product design including DSMs, MDMs and DMMs	Seeks to model and quantify any process model, either based on information flows or information dependencies	Seeks to quantify any process architecture either actual or planned	Allows for generic analysis at different levels of detail	Not explicitly	Allows for detailed structural analysis with focus on whole network level	Composition can be studied but is not detailed explicitly in the approach	Focus on binary matrices and direct paths but offers explicit approach to deal with indirect paths	Possible but not made explicit	Possible but not made explicit
Complexity Metrics in Engineering Design Processes (Kreimeyer & Lindemann 2011)	To quantitatively analyse the structure of process architectures to generate new insights	No but enables the quantification of network-based process model facilitating comparisons	No, but describes and quantifies the intersections between process architecture and the architectures of other domains	Process, product and organisation DSM plus MDMs and DMMs	Seeks to quantify any process model, either based on information flows or information dependencies	Seeks to quantify any process architecture either actual or planned	Architecture based on whole process data	Yes, for the whole process network	Allows for detailed structural analysis of whole process network	Only through the "number of classes" in a given network	Can be valued and consider indirect paths	Focus on identification of outliers based on structural metrics	Posed as a possibility through the analysis of the relationship between structural and performance metrics
Process Mining (van der Aalst 2011)	Generic approach to discover, evaluate and improve processes through a data-driven approach	Computed actual process can be compared against benchmarks based on planned or idealised processes	Not explicitly but it enables such a distinction through algorithms to discover the actual process	Event logs (dynamic) and activity list. Events need to be associated with one or more activities	Generic	Mainly actual	Whole process only	Intrinsically dynamic	It discovers the structure of the network and then allows to apply any structural metric	Can be computed at the whole process level	Valued and indirect paths are possible	Possible through, for example, the analysis of path frequencies	It allows to map performance on top of the structure

2.6 Current gaps

Although the architecture of the process domain depicted in section 2.3 is often used to describe and analyse actual information flows between activities, the relationship that connects activities is not an actual information flow. Instead, process models tend to map either information dependencies between activities (based on known technical and managerial needs), or intended and estimated information flows (typically in the form of top-down plans or perceptions of middle managers). A practical reason for this limitation is found in the direct mapping of activities to activities, which restricts the questions that can be posed to roughly two: 1) *What is the information dependency strength (if any) between activities A and B*? and 2) *What is believed to be or should be the information flow between activities A and B*? However, to model actual information flows between activities, we must consider the architecture of the multiple information exchanges among project participants in the context of specific activities. Those information exchanges constitute the actual information flow between any two activities in the process.

The distinction between a process model, built on information dependencies or planned and expected information flows, and a process model of actual information flows is important when interpreting certain research results. For example, the stated aim of Collins et al. (2010) and Braha and Bar-Yam (2007) was to describe and analyse the actual dynamics of information flows between activities; however, the information they acquired and modelled described only an evolving network of information dependencies, which in practice, limited their analyses to a technical view of the process domain. As a result, their conclusions should be restricted to the architecture of expected information dependencies, or if extrapolated, to planned or estimated information flows plans, rather than actual information flows.

For a more accurate, descriptive view of the process, data about information flows between activities should be based on the sum of actual information exchanges between people. As section 2.5 described, approaches at the intersection of the process and organisation domains have advanced in this direction, and approaches at the intersection of process and product architectures have pointed to the advantages of explicitly integrating cross-domain architectural information. However, the models and methods available to meet the open challenges and objectives defined in chapter 1 still have gaps. Those gaps demonstrate the need for a new approach that can extend the scope and contributions of existent models. More specifically, and in connection with the defined research objectives, the current gaps in the literature can be summarised as follows:

- **Characterisation** of the **actual engineering design process architecture** is insufficient.

  A characterisation of the actual design process requires approaches that connect process and organisation architectures with a process focus, allow for integration of the dynamic aspects of these domains and differentiation between structural and compositional aspects. On their own, none of the reviewed approaches completed fulfils these requirements.

- Existent models do not appropriately support a **comparison** between the **actual** and the **planned** design process architecture.

  The first step is to compare the actual against the planned design process to obtain a process that ideally is based on actual data. This comparison can be achieved using currently available approaches, but the results do not provide sufficiently accurate and flexible representations because of limitations of models of the actual process architecture.

- Current process models do not **characterise** the design process architecture at **multiple levels**, including activity, interfaces between activities, and the whole design process.

  No approach explicitly describes a systemic, multilevel characterisation of the actual design process in which the individual architectures of activities, interfaces, and the whole process are addressed and can be characterised.

- The means are limited to **connect the characterisation** of the design process architecture **with** process **performance** metrics to promote design process improvements.

  To fill this gap, a process model must at least consider and provide: a) a meaningful variability in the independent variable measuring the architecture, b) a dependent variable and/or benchmark, and c) a method to analyse the relationship between independent and dependent variables. No approach was found to include all these features.

Table 2-7 provides a summary of the reviewed approaches categorised by type of architecture, their temporality, the inclusion of people and/or activities, the possible comparison base or benchmark, and the main limitations for modelling actual design process architecture.

Table 2-7: Summary of current approaches to the architecture of process, organisation, and their intersections.

Architecture	Temporality	Examples	Inclusion of people or activities	Possible comparison base or benchmark	Main limitation to model the actual design process
Organisation	Static	Batallas & Yassine (2006); Hossain (2009); Kratzer, Gemuenden, & Lettl (2011); Sonnenwald (1996)	Only people	Can be compared against formal organisation architecture or in terms of cross-domain mirroring	Design activities or tasks are not considered
	Dynamic	Gopsill et al., (2014); Hossain, Murshed, & Uddin (2013); Kidane & Gloor (2007)	Only people	Does not count with a direct comparison base or benchmark	
Process	Dynamic (in the form of a sequence of activities)	Braha & Bar-Yam (2007); Browning (2002); Collins, Bradley, & Yassine (2010); Collins, Yassine, & Borgatti (2009); Smith & Eppinger (1997)	Only activities	Can be compared in terms of cross-domain mirroring	As people are not included, it cannot map directly the actual information flows between activities
Intersection Product-Process	Static with sequenced tasks	See table 2-4	Tasks and product parameters	Can be compared in terms of cross-domain mirroring or against the actual process architecture	As people are not included, it cannot map directly the actual information flows between activities
Intersection Organisation-Process	Static with sequenced activities	See table 2-5	People and activities with focus on organisation domain	Can be compared against information dependencies in the process domain	Limited to static views of the process; has an organisation domain focus; the multilevel nature of process architecture not fully addressed
	Dynamic	Christian (1995) (only simulation)	People and activities with focus on process	Can be compared against stages, information dependencies, and planned information flows	Developed as a simulation and does not characterise the multilevel nature of process architecture
Other cross-domain approaches	Dynamic and static	See table 2-6	Generic	Any benchmark or reference model	These approaches have many of the required features, but require adaptation to the specific intersection of process and organisation domains (focusing on actual process architecture)

2.7 Chapter summary

This literature review included the most relevant building blocks to develop a networked perspective on the engineering design process, at the intersection of process and organisation architectures. Section 2.1 set the overall research context, describing design as a socio-technical process of information transformation and introducing essential aspects about models and modelling applied to the design process. Section 2.2 examined what it means for something to be complex and showed concrete network-based approaches to analyse and make sense of complex socio-technical systems, such as the design of engineering systems. With the foundations provided in sections 2.1 and 2.2, the focus in section 2.3 turned to a particular view of the design process through the lenses of its architecture, reviewing network-based process models, particularly activity- and task-based models. Similarly, section 2.4 reviewed network-based models applied to the organisational and product domains. Section 2.5 presented network-based design process models at the intersection of product and organisation architectures, based on elements introduced in sections 2.3 and 2.4. Finally, section 2.6 integrated the key elements presented in the literature review and compared them to the research objectives to identify literature gaps and opportunities. This section guides the development of a framework for a networked perspective on the engineering design process (presented in chapter 4).

Key contributions of this chapter are the architecture-driven distinction between tasks and activities, and the presentation of various perspectives on process architecture, distinguishing among three ways in which process architecture has been conceptualised: 1) process architecture in which tasks or activities are directly connected through some implicit or explicit estimation about information dependencies (tasks) or information flows (activities); 2) the process at the intersection of product and process architectures in which information dependencies between tasks are explicitly modelled through the incorporation of interdependent component parameters; and 3) the process at the intersection of process and organisation architectures in which information flows between activities can be explicitly modelled through incorporation of people performing tasks and their mutual information exchanges. The combination of this information directly contributed to development of the framework presented in chapter 4.

3 RESEARCH METHODOLOGY

To ask the proper question is half of knowing
—Roger Bacon

This chapter describes the research methodology applied to this study and in particular to the development of the NPr Framework. The research methodology includes not only the system of methods employed to acquire, analyse, and interpret data, and how these methods are combined, but also the logic behind the selected methods in connection with the theoretical approach and its limitations (Blessing and Chakrabarti, 2009, p. 9). Because the developed framework is composed of a conceptual model—a set of analytical methods to quantify the model—and data-driven support, the methodological approach must consider and integrate all these aspects.

This chapter is structured as follows: section 3.1 provides a short review of theoretical and practical considerations related to the methodology and framework, section 3.2 elaborates on the previously introduced research objectives and questions, section 3.3 describes the overall stages following the adopted Design Research Methodology (DRM), and 3.4 describes the two industrial case studies and the strategies utilised for data gathering, analysis, and interpretation.

3.1 Theoretical and empirical approach

In dynamic socio-technical systems, such as the design process of an engineering system, data about the actual system architecture can be gathered directly through an examination of data and metadata generated by the digital objects and traces produced as part of the project's regular operations. This data includes email communication, activity logs, electronic documents, etc. (Giles, 2012; Hicks, 2013; Shi et al., 2014). Data of this type can be used to build models of the dynamic process architecture that include actual information exchanges between project members, actual participation in activities, actual process inputs and outputs, and other relational digital traces produced in the process. Such actual process data is essential to capture rich design patterns without straining the organisation members with continuous requests for information. At a more detailed level, data about individual thought processes and actions also is important, but often is not explicit in pre-existent data traces or digital objects, and therefore, is considerably more resource intensive to collect, compared with data about the process architecture.

Based on the previous discussion on complexity (2.3.1), we can assume that the larger and more complex the system, the bigger the scope for relational structures and the overall architecture to affect the whole system's behaviour. For example, the impact of a single individual in a large system will depend on the person's network embeddedness (Carpenter et al., 2012), and the more complex the network that the individual must navigate, the larger the influence the overall network can exert on each individual and the process (e.g. Chen & Gable 2013). For this reason and the practical access to data, I take the pragmatic approach and focus primarily on analysing manifest relational structures. Despite this main focus, this thesis also includes a non-relational aspect of the process architecture: system composition. Composition allows for incorporation of attributes associated directly with each individual, and offers a variable independent from the structure of individuals. Such non-structural variables can be used as a proxy for different types of individual behaviours, and thus, their impact on the outcomes of the design process can be tested individually and in combination with structure (e.g. Rodan & Galunic 2004).

Consistent with my decision to focus primarily on architecture instead of individual behaviour and thought processes, the methods used for data acquisition, analysis, and interpretation take a network science approach (Strogatz, 2001). The emphasis is on describing the network architectures that are produced by and affect individuals, and thereby influence the outcomes of design activities.

Research approach to develop the proposed framework

The developed framework represents this thesis's main contribution to knowledge and practice, and as a consequence, drives and explains the overall research approach I followed throughout this doctoral study. The process followed to develop the framework is summarised in the following sequence of steps:

1) Exploratory meetings with industry and initial literature review to identify industry needs and knowledge gaps to be addressed by the framework
2) Identification and organisation of the main industry needs and knowledge gaps
3) Initial definition of research objectives and research questions

4) Exploratory case study used as a first pilot to test new network approaches to design process architecture; concurrently, and until the end of the doctoral study, a second more focused literature review to continuously enrich the framework

5) Descriptive case study in which the full framework was applied and evaluated

6) Analysis of the results of the framework and evaluation of the framework's 'fit' to address industrial needs and knowledge gaps

Although this list appears as a linear sequence of steps, in practice, the actual research followed a series of iterative steps. Sections 3.3 and 3.4 in this chapter provide additional details about each of these steps.

3.2 Research objectives, research questions and success criteria

The research objectives and questions that guided this study, stated in section 1.2.1, were motivated by the needs and knowledge gaps identified in sections 1.1.3 and 2.7. In this section, I revisit the overall research aim and objectives, describe the research questions in additional detail, and connect them with expected outcomes and success criteria.

3.2.1 From needs and knowledge gaps to outcomes

The identified needs and knowledge gaps acted as the drivers of this thesis and were used to determine the goal of this research as well as concrete and feasible research objectives. The main unresolved issue for each of the three research objectives was phrased as a research question, which focused attention on the need and defined the shape of the expected outcomes. Finally, the outcomes—the results of the descriptive and prescriptive study—were evaluated based on academic and industrial success criteria. The set of success criteria was derived directly from the outcomes' capacity to respond appropriately to the original needs and knowledge gaps. Figure 3-1 depicts these relationships.

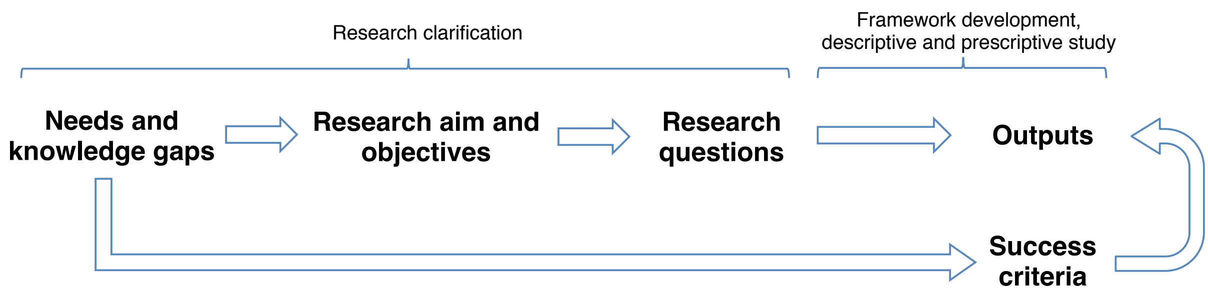


Figure 3-1: Relationships between needs and knowledge gaps, research objectives, questions, and outputs

I defined three interconnected research areas, from descriptive to the most prescriptive, based on exploratory work during research clarification and the research aim to **provide means to characterise the actual design process architecture and data-driven support to the design process of engineering systems**. These research areas, first introduced in section 1.1.3, were labelled conceptual characterisation of the actual design process architecture, quantitative characterisation of the design process architecture, and data-driven evidence and support.

a) Conceptual characterisation of the actual design process architecture

The first research objective was to develop a multilevel, dynamic characterisation of the actual engineering design process architecture. In terms of industry needs, this objective was triggered by an insufficient overview of the dynamic, actual design process and the fragmentation of current process models that do not include different levels of analysis in the same model. In terms of knowledge gaps, this objective stemmed from conceptual and analytical difficulties in characterising the architecture of the actual design process.

The main problem in this research area was translated into **research question 1: How can we model the multilevel, dynamic, and actual design process architecture of engineering systems?**

The answer to this first research question should provide a dynamic, multilevel **conceptual model** of the actual process architecture and a conceptual guide for the remainder of the research.

b) Quantitative characterisation of the design process architecture

The second research objective was to enable a comparison between the actual and the planned design process architecture. In terms of industry needs, this objective emerged from the difficulties that engineering companies face in comparing the planned design process with the actual process and its progress. In terms of knowledge gaps, this objective was triggered by constraints in the comparison of planned and actual engineering design processes, which is explained by two factors: 1) conceptual incompatibilities between the process views used to describe the planned process and those used to described the actual design activity, and 2) analytical limitations in quantifying in equivalent terms the actual and the planned design process architecture.

The main problem in this research area was translated into **research question 2: How can we quantitatively characterise the model of the actual architecture so that it is analytically comparable to the planned process architecture of engineering systems?**

The answer to the second research question should quantitatively operationalise the previously developed conceptual model (RQ1). For this, an **analytical method to quantify the conceptual model** had to be developed to enable quantification of the actual architecture at the three levels of analysis (activities, interfaces, and whole process) and make the model comparable with current models of the planned process architecture.

c) Data-driven evidence and support

The third and final research objective was to provide means for connecting the characterisation of the actual design process architecture with process performance metrics in order to support design process improvements. Because of the uniqueness of the industry, designed systems, and organisational characteristics, generic prescriptive advice has limited use, and therefore, this objective emerged from the industry's need for a way to diagnose the architecture of each design process directly. In terms of knowledge gaps, this objective addressed the need for sufficient variability to establish relationships between design process architecture and performance, which can allow for data-driven theory building.

The main problem in this research area was translated into **research question 3: How can we connect a quantitative characterisation of the actual architecture with process performance metrics?**

The answer to the third question should provide data-driven support to address each of the identified needs and knowledge gaps. The focus was on supporting design process improvements using the digital data traces that are already produced as part of the operation of the engineering design process.

3.2.2 Breakdown of research questions per level of analysis

Due to the multilevel nature of this research, which considers activities, interfaces, and the whole process architecture, each research question was broken down according to these three levels of analysis: activities (a), interfaces (i), and whole process (w). This division generated a 10 research sub-questions (RSQ), three per level in the case of activities and interfaces, and four at the whole process level. The whole process level required an additional sub-question so that the comparison between actual and planned process architecture could be included:

Sub-questions at the activity level:

i. How can we **model** the actual architecture of **activities**? (RSQ 1a)

ii. How can we **quantitatively characterise** the actual architecture of activities? (RSQ 2a)

iii. How can we test the relationship between the architecture of activities and their **performance**? (RSQ 3a)

Sub-questions at the interface level:

iv. How can we **model** the actual architecture of **interfaces** between activities? (RSQ 1i)

v. How can we **quantitatively characterise** the actual architecture of interfaces? (RSQ 2i)

vi. How can we test the relationship between the architecture of interfaces and their **performance**? (RSQ 3i)

Sub-questions at the whole process level:

vii. How can we **model** the actual architecture of the **whole design process**? (RSQ 1w)

viii. How can we **quantitatively characterise** the actual architecture of the **whole design process**? (RSQ 2w)

ix. How can we quantitatively **compare** the **actual** and the **planned** architecture of the **whole design process**? (RSQ 3w-1)

x. How can we test the relationship between the **dynamic architecture** of the **whole design process** and its **planned design stages**? (RSQ 3w-2)

A reason for the asymmetry between the levels of activities and interfaces and the whole process level is that a process includes many activities and interfaces between activities. Therefore, the characterisation of the architecture of activities and interfaces allows for analysis of the relationships between these many individual architectures in the process and their individual performance measures (or other features of interest). In contrast, only one architecture must be characterised at the whole process level—the whole process—and therefore, not enough variability is available to analyse relationships between different types of architectures and performance. To respond to the limitation that this represents for analysis of the relationship between whole process architecture and performance, I considered two strategies, translated into sub-questions:

1) To analyse the actual process architecture side-by-side against the planned process architecture and compare the extent of the alignment between the actual and the planned process (sub-question ix)
2) To analyse the whole process architecture dynamically so its evolution can be benchmarked against planned or prescribed design stages (sub-question x).

3.2.3 Success criteria

Based on the research objectives and questions, I used the following measurable criteria as a guide to evaluate the success of the proposed framework in industrial and academic terms:

Industrial success criteria

- Conceptual model: The framework should deliver an improved overview of the actual design process through a model that achieves face validity from the company's perspective. The model should be operationalisable, making use of information that is economically feasible to gather and representative of the process.
- Analytical method: The quantitative characterisation of the actual design process architecture should provide companies with the practical and intuitive means to map their actual design processes and compare them against their planned processes.
- Data-driven support: As a result of the developed framework, the case studies should report increased awareness and improved understanding about their actual design processes, relationships between process architecture characteristics and process performance metrics, and the differences between their planned and their actual process architectures.

Academic success criteria

- Conceptual model: The framework should develop a model of the actual design process architecture that brings new insights into the actual process architecture when compared with previously available approaches. The model should address the dynamic and multilevel nature of the design process architecture.
- Analytical method: The framework should provide a quantitative characterisation of the developed model. This characterisation should be comparable to planned design process models and integrate all the levels of analysis defined by the conceptual model.
- Data-driven support: The developed framework (conceptual model plus analytical methods) should provide a flexible and quantitative platform for future research seeking to identify relationships between actual process architecture and process performance metrics.

3.2.4 Overall organisation of research questions, sub-questions and outcomes

Chapters 4 and 5 are organised based on the main research questions and the three levels of analysis. Figure 3-2 provides a visual summary of the main sections in which the outcomes for each research question and sub-question are introduced. The answers to RQ3 (How can we connect a quantitative characterisation of the actual architecture to process performance metrics?) and its sub-questions are integrated as key considerations throughout the development of the conceptual model and analytical method.

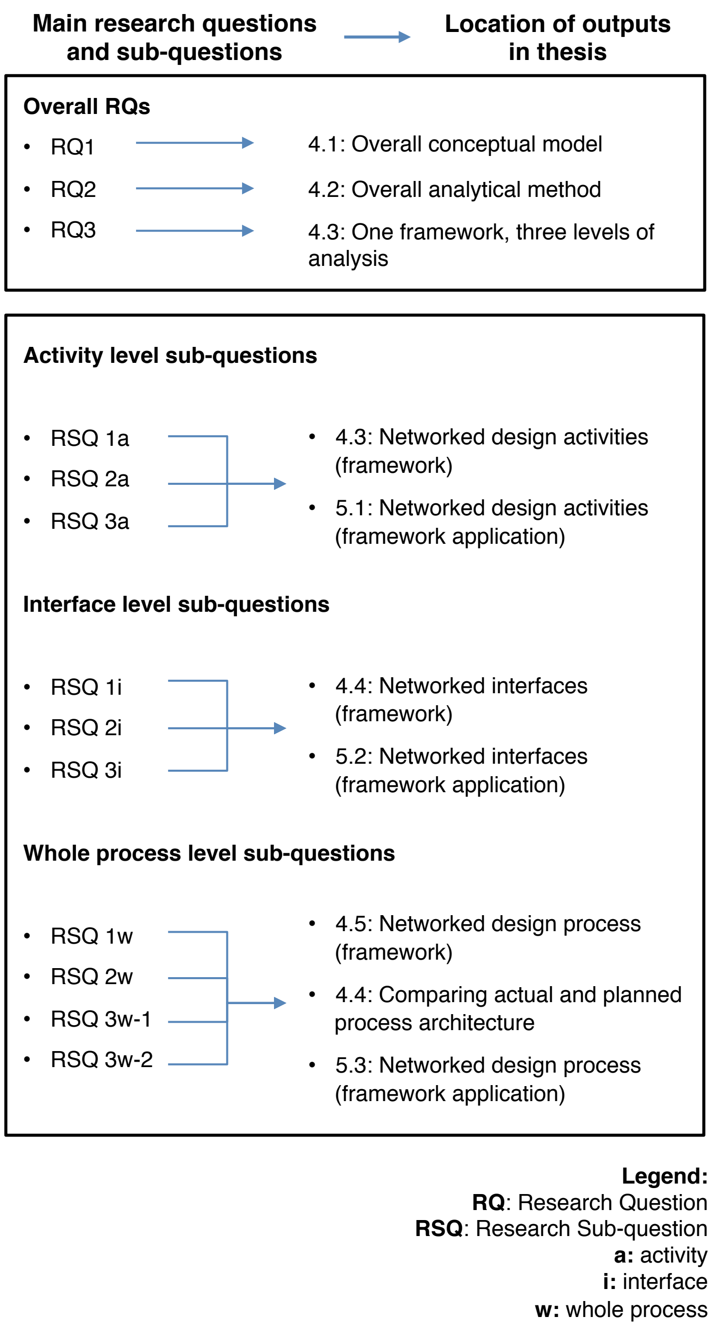


Figure 3-2: Visual summary of the location of each research question and sub-question outputs

3.3 Design research methodology stages

For this thesis, I used Blessing and Chakrabarti's (2009) design research methodology (DRM) as an overall guide to develop descriptive and prescriptive contributions to design theory and practice.

The DRM consists of four stages, which I followed to structure this thesis and guide the research process: research clarification (chapters 1–3), descriptive study I (chapters 4–5), prescriptive study (chapters 5–6), and an initial stage of descriptive study II (chapter 6). The first three stages, shown in figure 3-2, were associated with at least one research question. The last stage, descriptive study II, was focused on 'the impact of the support and its ability to realise the desired situation' (Blessing and Chakrabarti, 2009, p. 16), and therefore, this stage focused on an evaluation of whether the success criteria were met.

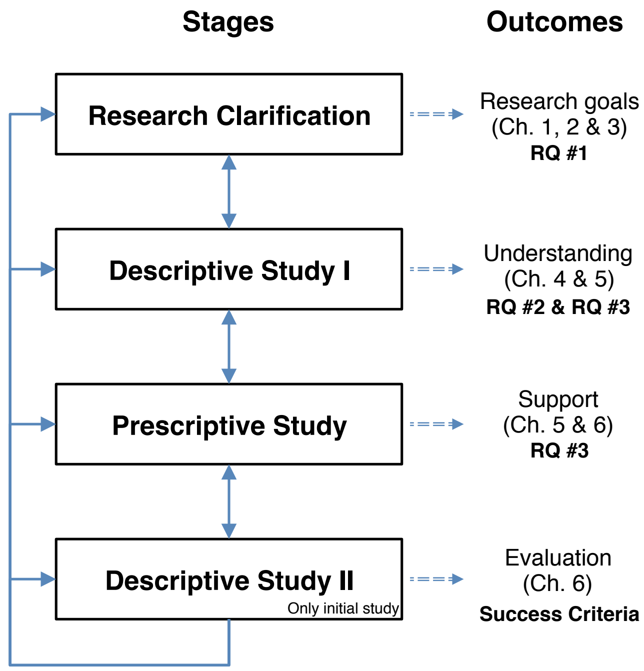


Figure 3-3: Research stages in this thesis following the design research methodology

The goals and the work performed at each stage are summarised as follows:

Research clarification

The goal of this stage was to define the key research problems, research objectives, theoretical focus, research questions, and to identify potential models and methods to answer the research questions. The main work performed included a literature review focused on engineering design, complex socio-technical systems, and network science. To frame the problem and identify the most pressing unaddressed issues, I conducted short visits to five medium and large Danish companies with in-house engineering, attended an industrial conference on new cleantech product development, and conducted several open and semi-structured interviews with practitioners in the visited companies and at the industrial conference.

This stage included an inductive process of increasing abstraction, whereby needs discovered in industrial practice and literature gaps were framed into approachable research objectives and connected to a research methodology. The intention was to provide a holistic exploration of the problem space that subsequently could be transferred to a suitable model of the design process. The direct results of this stage are found primarily in chapters 1–3.

Descriptive study I

The goal of this stage was to develop, refine, and empirical test the conceptual model and analytical method for characterising the actual architecture of the design process. The main work at this stage was the iterative development and application of the framework with the two case studies, combined with a second literature review, which was necessary to provide the framework with the required theoretical grounding.

The exploratory case study, based on a project to develop a flat-sheet membrane for water filtration, was used to examine various research approaches and as an early pilot for the developed framework. The main research methods in this first study included two weeks' of observations, semi-structured interviews, structured interviews, and a document analysis of company files (see more details in appendices). From this case, I developed an initial working version of the model and method that was later tested on a large-scale engineering design project.

The second case, the engineering design of a biomass power plant, was approximately 10 times larger that the exploratory case, both in numbers of people involved and coded activities. The main research methods in this second case included one week of observations, semi-structured interviews, electronic questionnaires, and the elicitation of detailed company datasets, including event and activity logs and internal models of the design process (see more details in appendices). I used this second descriptive case to apply the final version of the framework, and to analyse and interpret its results.

The work during this stage involved a deductive process of increasing decomposition: Through the empirical analysis of multiple independent and dependent variables, the findings were further elaborated and divided on more approachable analytical components. Most of the research methods applied at this stage were quantitative; however, at the beginning of this stage, a qualitative exploration of the two cases studies was required to set the organisational and technical context in which to apply the quantitative analysis. The main results of this stage, the developed framework and its application, are found in chapters 4 and 5. This stage primarily addressed research question 2, because the objective was to develop and refine the method for quantifying the model through iterative work with the case studies.

Prescriptive study

This stage's objective was to use the quantitative characterisation of the actual process architecture produced in the previous stages to develop concrete means to support design process improvements and generate new knowledge. In this thesis, the levers to transform quantitative characterisation of the actual design process architecture into value for industry and academia are the relationships between the actual process and performance metrics, the planned process architecture, and the design process stages. Therefore, I focused on developing design support able to connect process architecture with variables of performance or other benchmarks to permit interpretation of design patterns, making those patterns meaningful for an enhanced process overview, decision making, and theoretical insights.

The development of design process support was iterative through interviews and presentations with the participating case studies, in which ideas, various visualisations, process models, and quantitative results were shared and refined to accommodate industrial needs and represent reality. The goal was to identify the most useful findings, so efforts were prioritised based on the design support that most efficiently and effectively improved the process overview and supported decision-making in the design process.

The prescriptive study was performed on the two case studies; however, the first case included only an initial prescriptive study, while the second case had a comprehensive one. This stage utilised an inductive process of knowledge integration, whereby a combination of qualitative and quantitative approaches were used to transform the most promissory findings into design support applicable beyond the boundaries of the case studies. Chapter 5 includes the main results of this stage and detailed explanation of the framework's application. This stage primarily addressed research question 3.

Descriptive study II

The objective of this stage was to perform an initial evaluation of the support developed during the prescriptive study. My strategy was to qualitatively assess whether the support improved the company's design process in descriptive case study. The main inputs at this stage were follow-up interviews and presentations at the company followed by company feedback. However, the time between the first prescriptive advice and the end of this doctoral research did not permit a comprehensive evaluation, and feedback was limited to broad information about the degree of knowledge absorption and internalisation of new practices.

Although the previous stages were defined as 'comprehensive studies', this research stage fits only the definition of an initial study: 'An **initial study** closes a project and involves the first few steps of a particular stage to show the consequences of the results and prepare the results for use by others' (Blessing and Chakrabarti, 2009, p. 18). As a result, this stage's objective was to show overall and preliminary findings and prepare the results for others to use to pursue additional studies in this line.

Although the descriptive study II was performed mainly on the descriptive case study, I also received unstructured feedback from the exploratory case study. In this stage, I employed a qualitative process of assessment, based on the company's perceptions of the provided design support. The results can be found in chapter 6 through the evaluation of the defined success criteria.

Figure 3-4 provides a graphic summary of all four stages applied to this research.

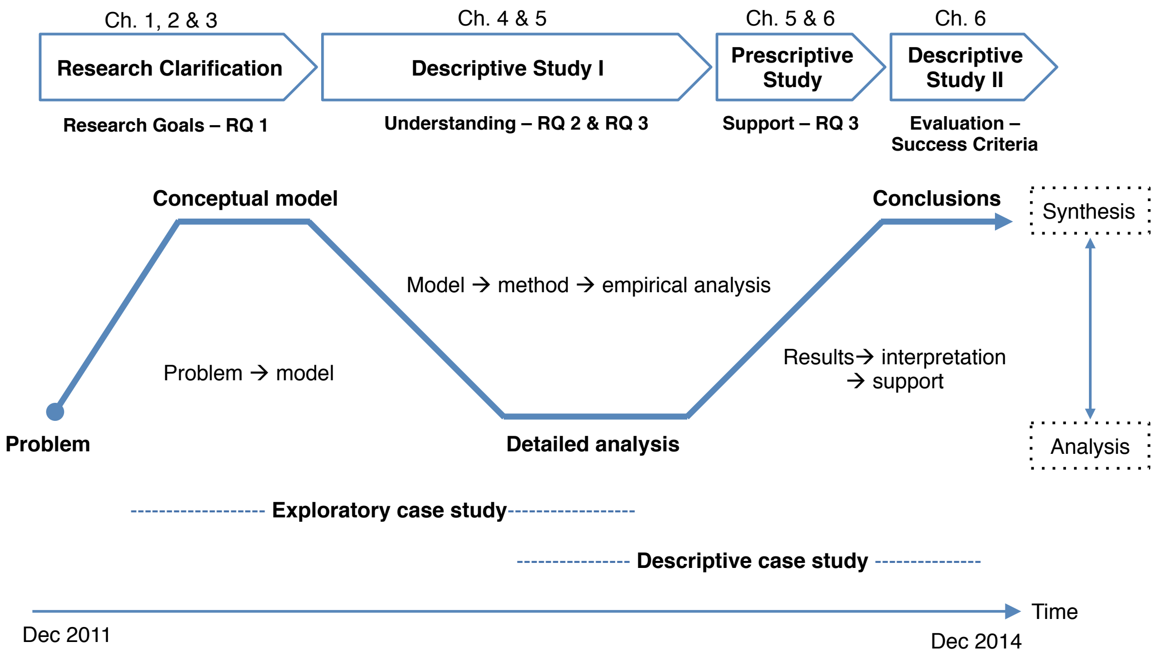


Figure 3-4: Design research stages applied to this thesis

3.4 Empirical studies

This section describes the two empirical studies used to iteratively develop and test the model, the quantitative method, and the data-driven support to the design process of engineering systems. The section is includes first a description of the design and objectives followed in the case studies (3.4.1), and second, a description of the two case studies through a company profile (3.4.2). Finally, I offer details about the research methods utilised to acquire, analyse, and interpret the case study data (3.4.3).

3.4.1 Case studies design and objectives

Although the three research questions could have been answered without case studies, relying instead exclusively on extant literature, assumptions, and secondary and/or simulated data, the research's validity would have been more difficult to determine. Such a disconnection from the industry also would have imposed a higher risk, because a tight relationship with real industrial needs and practices was essential for the success of the developed design support (Blessing and Chakrabarti, 2009, p. vii).

However, I did not use the case studies for the purpose of theory building or direct theory testing, and therefore, did not encounter the same specific challenges and requirements identified in Eisenhardt and Graebner (2007) for case studies employed for such purposes. Rather, I used the cases as an instrument for research clarification, iterative development of design support, proof-of-concept for the developed framework, and evaluation of the success criteria.

Following Yin's (2009, chap.2) classification of case studies, the two case studies in this thesis are categorized as a first exploratory case study and a second descriptive case study. Case study 1 was an exploratory case study and a pilot for testing preliminary ideas, developing a better understanding of the research area, defining actionable and relevant research objectives, and developing the framework (Yin, 2009, p. 78). In contrast, the descriptive case study 2 was used to apply and test the developed framework, working as a full-scale proof-of-concept. Although two case studies were used, the research design does not fit the classification of 'multiple case studies' (Eisenhardt and Graebner, 2007; Yin, 2009, p. 33), because the objective was not replication, but iterative refinement of the models and methods, moving from an exploratory and pilot case study to a descriptive one.

Selection of the case studies

At the methodological level, the main selection issues were the selection of the organisations and the projects.

Organisation selection: Based on research needs, I determined that two cases were necessary: one an exploratory and pilot case study to clarify the research questions and further develop the model and method, and one a descriptive case to perform a full-scale application.

To improve generalisation, the two case studies needed to come from different organisations that develop different types of technologies.

With this background, these selection criteria were defined: (A) The companies selected should have at least one complex engineering design project recently finished or with a due date during the course of this research. This requirement allowed recent data for the entirety of the selected project's lifetime was available and information could be collected directly from most of the project's participants. (B) Because data about the information exchanges between participants would originate from the focal companies where the case studies were based, the selected companies should lead the engineering design activities of their respective projects. (C) For practical reasons, the selected companies' main engineering design operations should be in Denmark. Once these criteria were satisfied, the selection process was opportunistic, based on the companies' positive predisposition and willingness to participate.

I selected the two required cases after meeting with industry associations, innovation networks, and conducting more than 10 interviews with personnel from potential companies that met the required criteria and showed interest.

Project selection: All the qualified companies also had two or more suitable projects; therefore, a set of criteria was necessary to select between the projects in their portfolios. These criteria were: (A) recently finished or nearly finished project; (B) high technological, organisational, and/or process complexity; (C) active cross-functional and inter-organisational involvement.

Goals pursued through the case studies

By using case studies, I sought to fulfil certain concrete goals, which are described according to relevancy to the exploratory or the descriptive case study:

Exploratory case study goals

- To identify, understand, and prioritise unsolved industrial needs in order to align research efforts with real industrial needs
- To identify and design realistic strategies for data acquisition that take into consideration data availability and quality, thus minimising data acquisition costs so that companies are more likely to implement the data-driven support
- To adapt the model and method in order to utilise the discovered realistic data acquisition strategies

Descriptive case study goals

- To iteratively test the model and method with real data
- To interpret and evaluate the results obtained from the empirical application of the model and method with design process participants to confirm their consistency with the real design process
- To evaluate the relevance and usefulness of the developed support through direct application

These empirical research goals are intrinsically connected with the three main overall research objectives (sections 1.2 and 3.2).

3.4.2 Case study descriptions

The two empirical case studies differed in terms of company size and industry. The exploratory case study involved the design process for a flat-sheet membrane for water filtration, and the descriptive case the design process for a biomass power plant. This section provides the facts about each of these cases to give context for the results obtained from the application of the framework on the descriptive case study.

Exploratory case study: the design process of a flat-sheet membrane

The exploratory case study was carried out in a Danish company that designed and manufactures silicon carbide ceramic filters for gas and water applications. Founded in 2001, the company had in-house R&D and manufacturing capabilities, employed approximately 100 people, and had commercial operations in Denmark, the United States, Singapore, Germany, France, and Korea. The R&D process occurred in one location in Denmark, where most of the manufacturing process was collocated. Currently, the company is under re-organisation following merger and acquisitions processes.

The selected project was the development of a new kind of filter, a ‘silicon carbide flat sheet membrane’ (SiC FSM), to complement the company’s previous line of tubular filters, particulate, filter technology, auto catalyst, and kiln furniture. SiC FSMs, which are used in biological bioreactors for water filtering and have significant market growth potential, represented the company’s most important new product development project during that time period. The development process started with a conceptual proposal around September 2011 and ended in May 2013.

The design process included the design of the SiC FSM itself, as well as the processes and tools required to take it to production. The design of the SiC FSM was an iterative process involving engineers from R&D, sales, and manufacturing. I interviewed the 10 engineers directly involved in the project, and they mentioned eight other participants outside the core engineering team (in other departments and organisations). These persons were subsequently included in the project network but not interviewed. The design process had a team of five members affiliated with the R&D department, four with sales, and one with manufacturing. Despite their formal organisational affiliations, all engineers where part of the same design process for this project. The project leadership was in the hands of the company's chief engineer, and the executive coordination was through a project manager. The process architecture included eight work packages, each assigned to one activity group, with a total of 38 activities among the eight activity groups.

Descriptive case study: the design process of a biomass power plant

The descriptive case study involved a Danish company whose main business is the engineering design of power plants operating boiler-based technology. The company is owned by a large international holding operating in the heavy industry sector. Founded in 1843, the focal company has more than 130 active employees, global sales, and engineering design functions primarily in one location in Denmark. Although it does not manufacture directly, the company has a strong relationship with an international network of suppliers and manufacturers with whom it coordinates the procurement process. Given the complexity of the technology and the need to rapidly identify and correct integration problems, the company also actively follows the on-site building process for the power plants.

The selected project was the engineering design process for a biomass power plant, performed in coordination with a partner company and a network of more than 56 external organisations. The engineering design work was carried out between September 2009 and August 2013. I gained access to project data through the company in charge of the project's engineering design and the same focal company coordinated the work with the construction partner, manufacturers, and components providers. Key contacts included the vice president of operations, the vice president of engineering, the project manager, and the quality assurance team, all of whom were interviewed. The design process architecture included 13 work packages, each assigned to one activity group, with a total of 148 activities among the 13 activity groups.

3.4.3 Case study research methods

The main case study methods depended on the research stage, the case study, and the case objectives. During research clarification, the focus was on open and semi-structured interviews, company visits, and a review of company documents. During the descriptive study, I focused was on structured company databases and other digital traces, as well as semi-structured interviews and in-company observation. During the prescriptive study, my focus was on utilising the quantitative characterisation of the actual architecture and developing appropriate support through interviews. During the descriptive study II, I used interviews to obtain initial feedback about the appropriateness and usefulness of the provided support. These main data acquisition and research analysis methods are discussed in the next sections.

Data acquisition

Quantitative data acquisition

The main goal of the quantitative data acquisition was to refine and test the proposed model and method, enabling a full proof-of-concept of the intended design process support.

The quantitative data acquisition was operationalized through:

- **Pre-existent structured process data**: Companies often record information traces that can be used for traceability purposes, information management, budgeting, process analysis, etc. This research focused on the design process's network structure and composition, meaning that databases and other company documentation revealing interactions between people, workflow diagrams, and activity logging of the actual design process held particular interest.

  Advantages of using already existent data sources include the minimisation of distortions that the researcher's data acquisition efforts may introduce to the targeted organisation (Van de Ven, 2011). Existent sources also provide a more replicable and scalable stream of information because data production to feed the research does not impose an additional cost on the organisation (Allen et al., 2011). The main disadvantage is a loss of control of the type and quality of data received as inputs. I compensated for this disadvantage with a thorough understanding of the actual characteristics of the database, including data errors, incompleteness, and biases. For example, in collaboration with the company, activity codes found in databases where manually cleaned, organised into meaningful categories, grouped, and recoded through a qualitative data acquisition process. The company databases used in this research included time-tracking systems linking activities and people as well as document management and exchange systems.

- **Electronic questionnaire:** To complete information only partially available on existent databases and acquire information for which explicit records did not exist, I developed a closed-answer electronic questionnaire, which was administered to all members of the core group in charge of the engineering design process. The questionnaire (see appendix B) asked individuals to report only those aspects directly connected to themselves and their functions, and included questions in these areas:
    - Departmental affiliation (pre-filled alternatives)
    - Participation on design activities (pre-filled alternatives)
    - Assessment of participation in design activities, including perceived responsibility, perceived activity's resource efficiency, perceived activity's result quality, perceived activity's innovativeness, and perceived overall performance
    - Process-related information exchanges with other project members inside the focal company (pre-filled alternatives with the possibility of specifying additional names)
    - Process-related information exchanges with external organisations (pre-filled alternatives with the possibility of specifying additional organisations)
    - Personal assessment of the information exchanges based on impact of the interactions in the performed activities (low, medium, or high), the interactions' directionalities (initiated, mutual, or received), and the interactions' frequency (daily, weekly, or monthly)
    - Personal assessment of the project, including performance (efficiency, quality, innovativeness, overall), personal project knowledge, and satisfaction with the personal knowledge and overview of the project

The questionnaire was developed following a whole-network, weighted and undirected approach for bimodal networks (Borgatti et al., 2013). The key relational structures were people-people information exchanges and people-activity affiliations, both of which were considered independent variables in the model. In addition, performance attributes for people and activities served as dependent variables. The operational boundary was the lifetime of the engineering design project. The goal was to include all activities and all people in the focal organisation (the lead organisation and site of the field study) as well as information about contact points with external organisations in the context of the project. Not all people were available to complete the questionnaire; some had left the project and were not reachable. To complete the whole network map, I symmetrised the available information based on the assumption of reciprocity (Borgatti et al., 2013; Wasserman and Faust, 1994), which in combination with the robustness of network data against minor data incompleteness (random and below 20%) provided a good base for analysis (Borgatti et al., 2006; Wang et al., 2012).

Qualitative Data Acquisition

The qualitative data acquisition methods had the following objectives: setting the context, boundaries, and scope for each case study; identifying, assessing, prioritising, and if necessary completing or modifying quantitative sources of information; and, refining and validating the quantitative instruments before they were applied. All qualitative methods feel within the category of field study because they implemented in direct contact with the case study company and were non-experimental.

The qualitative data acquisition was operationalised through:

- **Direct observation:** Each case study included an in-company observation period, combined with the study of company documentation and face-to-face interviews. The purposed of the observation period was to establish a grounded understanding of the engineering design project, the organisational context, and its potential influences on the design process. The observation was performed during the early stages of this research to minimise any interference with the organisation's normal operations that could bias or condition the receptivity of results and proposed support (Yin, 2009, chap. 4). During the observation period, field notes were kept in a journal, and whenever possible, photos were taken and annotated.
- **Study of company documentation:** To complement data from the observation period, I studied unstructured public and private company documentation, including public records available online such as the company's history, product lines, and patents, and private documents such as project timelines, workflow diagrams, organisational structure, and other project management records. Records with direct influence over the project design process that required clarification were structured and used to develop follow-up interview questions.

- **Face-to-face interviews:** A range of semi-structured interviews were conducted during the various research stages (Abbott, 2013, p. 206). These were held before and after each round of quantitative data acquisition in order to identify, assess, prioritise, and if necessary, complete or modify the quantitative data-gathering instruments or the quantitative information itself. Interviews ranged in length between 30 and 90 minutes and included project participants directly involved in the design process. Most interviews were with senior project members, including project managers, the chief of engineering, and the vice president of operations. (See appendix C for a detailed list of interviewees for each case study.).
- **Interactive presentations of research results:** As an additional source of data, key project members were presented with an initial interpretation of the results after each round of data analysis in order to gather their feedback and ground the data interpretation.

Data Analysis

Quantitative data analysis

For the quantitative data analysis, the independent variables were the compositional and structural network characteristics, and the dependent variables were the performance attributes at the activity and project levels. The first step was to utilise network analysis techniques to reach a systematic and quantitative characterisation of the target network architectures. The second step was use inferential statistical analysis to establish relationships between the independent and dependent variables.

The quantitative data analysis employed the following:

- **Network analysis:** The main software packages used during this research were *UCINET 6* for analytical calculations (Borgatti et al., 2002), *Gephi 0.8.2-beta* for visualisations (Bastian et al., 2009), and *Condor 2* (Gloor, 2013) for dynamic network analysis. Key network metrics were size, density, betweenness centrality (Freeman, 1977), group betweenness centrality (Wasserman and Faust, 1994), and information centrality (Stephenson and Zelen, 1989). A set of macros for Excel and for software running RStudio and Shiny applications was developed and bundled as a software suit called Net-Sights (see appendix G). Details about the selection of these network measures and their implementations are provided in chapters 2 and 4.

- **Statistical analysis:** The main software package for descriptive and inferential statistics utilised during this research was SPSS 20 (IBM Corp, 2011). Key analysis included two-step clustering (IBM Corp, 2001), one-way ANOVA, and linear regression analysis. Details about the selection of these statistical analyses and their implementations are provided in chapter 5.

Qualitative data analysis

Qualitative triangulation against company documentation and the semi-structured, face-to-face interviews was performed during the descriptive study I stage to verify the findings. The objective was to identify errors or significant discrepancies between the quantitative data analysis results and the qualitative inputs from the field study. Any identified errors or discrepancies were amended or taken into consideration before interpreting the results.

During the prescriptive study, I conducted a qualitative data analysis in conjunction with members of the case study organisation to analyse results obtained during the descriptive study I. This analysis guided the subsequent data interpretation of the prescriptive study and the descriptive study II.

Table 3-1 provides an overview of key facts related to the two case studies and the main case study methods employed.

Table 3-1: Overview of key facts and case study methods per case study

Key facts		**Exploratory case study**	**Descriptive case study**
Basic information about the studied design process (Includes only the engineering design project at the focal company)	# People involved in the engineering design project	10 core team members / 18 people, considering the extended team	49 core team / 96 people, considering the extended team
	# Departments or areas involved	3 internal departments: sales, R&D, and manufacturing	15 areas divided by engineering function
	Process breakdown	8 work packages, each assigned to one activity group. A total of 38 activities were coded among the 8 groups.	13 work packages, assigned to one activity group each. A total of 148 activities were coded among the 13 groups.
	Project lifetime (data was captured for the entirety of each design process)	September 2011–May 2013	September 2009–August 2013
Main research stages	Research clarification	Comprehensive	Partial
	Descriptive study I	Partial	Comprehensive
	Prescriptive study	Partial	Comprehensive
	Descriptive study II	Initial	Initial
Case study methods	# Weeks of in-company observations	2	1
	Open and semi-structured interviews (See appendix D for interview guides)	14	7
	Structured interviews	10 (See appendix D)	0
	Electronic questionnaire	0	49 fully completed questionnaires (See appendix B)
	Key secondary information and database information	• Gantt charts • Project plans and logs • Project technical assessments	• Workflow diagrams • Activity logs • Document management system logs • Human resources databases with affiliations and organisational charts

3.5 Chapter summary

This chapter covered theoretical and practical considerations related to key methodological choices, particularly the rationale behind my focus on manifest process architecture instead of other types of data. Second, this chapter detailed the research objective, questions, and the success criteria that narrowed the scope and organised the research. Third, I described the design research methodology stages and linked them to chapters in this thesis; outlined the research questions, and provided an overall timeline of the research process. Finally, this chapter provided information about the two case studies and the specific methods used during development and application of the framework.

4 THE NETWORKED PROCESS FRAMEWORK

The game of science is, in principle, without end. He who decides one day that scientific statements do not call for any further test, and that they can be regarded as finally verified, retires from the game

—Karl Popper

This chapter describes the networked framework developed to characterise the actual design process architecture and to support the design process of engineering systems. The framework consists of three elements: **a conceptual model**, **a set of analytical methods** to quantify the model, and **data-driven support** for the engineering design process. Together, these three elements provide a networked perspective on the engineering design process, a perspective at the intersection of the process and organisation architectures. For simplicity, the analytical methods and the data-driven support are presented as a single unit whenever the developed analytical method is used at the same time as the data-driven support mechanism for industry. To emphasise the networked nature of the framework, I have named it the 'Networked Process Framework', or for short, the '**NPr Framework**'.

Conceptual model

The model developed in this thesis builds on and enriches current activity network models, such as process DSMs and workflow diagrams, by explicitly integrating new aspects and details to existent descriptions of the design process architecture. These aspects include:

- The inclusion of **actual information exchanges between people** in the process and their participation in activities
- A distinction between **structure (**arrangement of and relationships between the elements of the analysed system) **and composition** (the elements of the system and their attributes) when characterising the actual process architecture
- **Three well-defined levels of analysis:** activities, interfaces between activities, and the whole activity network. Each level is individually characterised in terms of its own architecture, which is divided according to network structure and composition.
- An **integration of the three levels of analysis**, allowing **comparison of the actual process architecture against the planned process architecture and performance metrics**

- A more detailed and flexible approach to the **temporal dynamics** of the actual process architecture

This enriched process model is a better reflection of the actual information flows between activities. The model's results can be represented using traditional process model views, such as the ones based on information dependencies between tasks, and therefore, the model can be used for comparisons. The model also provides the means to connect the characterisation of the design process with process performance metrics because its multilevel nature generates the necessary variability in the independent variables that measure network structure and composition.

The model's relationship and complementarity with previous process models, as well as its distinctive contribution to theory and practice, can be found in chapter 5 (applied results per level of analysis) and chapter 6 (overall model evaluation and discussion).

Analytical method

The analytical method use a combination of quantitative analysis techniques to measure the conceptual model and support data-driven decision making, visualisation, and reflection. Some of the key elements of the proposed analytical method include:

1. **Network analysis,** including nearness matrices, structural measures for centralisation at the ego-network, and whole network levels comprising both static and dynamic metrics
2. A combination of **network visualisations** and **charts** to reveal both the aggregated (static) network structure and the evolution of key network metrics
3. **Statistical methods** to cluster and test for meaningful differences between groups of activities or interfaces
4. **Metrics** to quantify **compositional variation**

Although these quantitative methods carry their own analytical limitations, the same model can be implemented through different quantitative methods that may be more suitable for other research contexts and applications. In addition, the model and the analytical methods themselves can be modified or extended. Such potential adaptations allow addressing originally unforeseen requirements, incorporating new and improved methods , or responding to changes in the underlying assumptions. For this reason, I treat the model and method separately in the remainder of this chapter. Then, in chapter 5, I apply the framework as a whole to the descriptive case study. This application integrates the developed conceptual model, analytical method, and data-driven support to provide a complete proof-of-concept and to demonstrate the capabilities of the NPr Framework.

4.1 A conceptual model of the actual design process architecture

Sections 1.3 (research scope and assumptions) and 2.2.1 (network-based methods) introduced a key assumption that provides the rationale for the network-based modelling approach developed in this thesis. The assumption is that the architecture of the design process (structure and composition) affects the behaviour and ultimately the performance of the process of designing. This assumption follows not only the reasoning of the Function-Behaviour-Structure (FBS) ontology proposed by Gero and Kannengiesser (2002), and particularly its extension from objects to processes (Gero and Kannengiesser, 2007), but also the more general consensus in the study of complex systems, which is that the architecture of a system is one of the main drivers of the systems' behaviours (Davidsen, 1992; Gordon, 2014; Johnson, 1992).

If such a relationship between structure and observable behaviour is also true for design process architecture—and previous studies have shown empirical evidence to that effect (e.g. Browning & Eppinger 2002; Yassine, Joglekar, et al. 2003; Wynn et al. 2014)—then it is essential to obtain an accurate description of the process architecture, given economically feasible data sources and current means for analysis. The better the description, the more we can do to map and predict the relationship between particular configurations of the architecture and desirable or undesirable behaviours. This detailed information is especially relevant for behaviours that directly affect performance metrics having to do with timelines, budgets, and specifications.

There is an additional and important reason to invest in more precise models of the actual design process: Participants in complex engineering design processes need an overview not only of the engineering design project based on the planned sequence of activities (centred on expected information dependencies), but also of what actually happens so they can exercise reflection-in-action (Schön, 1984). **Such an active comparison between plans and the complex observed reality lies at the centre of the feedback mechanism that enables learning, and thus, design process improvements** (Busby, 1998; Sterman, 1994).

But, what makes the actual process architecture more difficult to model than the architectures of the product or organisation domains? Although there are practical challenges in modelling product and organisation architectures, they appear to be more transparent than process architecture. For example, the components of a product can be observed and listed in precise detail, and their multiple interactions can be quantified. Similarly, participants in an organisation can be listed, observed directly, and their actual relationships and interactions captured and quantified. Although there are practical and economical limitations to describing these two architectures, especially within complex engineering design projects, their fundamental units of analysis (components and people) are directly observable, which is not the case with activities and tasks in a process.

Following from the previous considerations, design process architecture is more difficult to model because it belongs conceptually to a different kind of architecture, one in which elements cannot be mapped directly to one another (as in product and organisation architectures), but must be mapped indirectly. Indirect mapping is required because activities are a complex construct that cannot be observed or measured directly as a unique entity. Consequently, models that start with direct mapping of activities tend to introduce a higher degree of subjectivity, have practical limitations in aggregating more granular information sources, and complicate separating the planned from the actual architecture. Therefore, process models based on direct mapping, such as Collins et al. (2010) and Braha and Bar-Yam (2007), face conceptual problems when attempting to model actual information flows (this situation was explored in section 2.6, 'current gaps').

A modelling framework that addresses the process domain's distinct nature, and the related need to map connections between tasks through observable entities, is the parameter-driven model of 'Signposting' (Clarkson and Hamilton, 2000; Wynn et al., 2006). Despite the completeness and power of the Signposting Framework, its scope is primarily the technical angle of the design process, at the intersection of the process and product architectures. Such process architecture can be used to characterise information dependencies between tasks and to provide essential input to design and manage the design process; however, **the activities of the actual process architecture** have to do with **dynamic information exchanges through interconnected people performing activities, and not interdependent design parameters and information dependencies.**

Although a rational design process should be expected to follow the logic sequence of information dependencies between tasks, in reality this might not be the case. For example, there may be unwanted organisational silos, power struggles between project participants, lack of coordination, unexpected constraints, changes in the environment, and last but not least, the original plans may not have considered important technical aspects and innovation opportunities that are discoverable only during the actual design process. Although this social and dynamic aspect of designing is addressed in management and engineering design literature (e.g. Drucker 1998; Bucciarelli 1988; Pahl et al. 2007, p.9), it is not easily integrated into the technical dimension of process architecture that frameworks such as signposting portray. One objective of this thesis (research question 2) is to make these aspects comparable so that a systemic socio-technical characterisation of the design process can emerge.

In summary, to derive a more accurate description of the actual design process, we must examine the intersection of process and organisation architectures, which requires connecting activities indirectly through those who perform them and through the network of information exchanges between project participants. Although other approaches at the intersection of process and organisation architectures exist (section 2.5.2), they work predominantly in the organisation domain and have other limitations and gaps (see section 2.6).

A new approach to describe and analyse the design of engineering systems should therefore represent design as a social process of information transformation. Figure 4-1 depicts this rationale, illustrating the product and organisation architectures and the mapping between product components and tasks, as well as the mapping between people and activities. The figure shows that the actual process architecture can be accessed by:

- Combining the product architecture with the mapping of components to tasks, which results in an actual process architecture based on 'technical' tasks (the route taken by approaches like the Signposting Framework)
- Combining the organisation architecture and the mapping of activities to people, which results in an actual process architecture based on 'social' activities, the focus of this thesis.

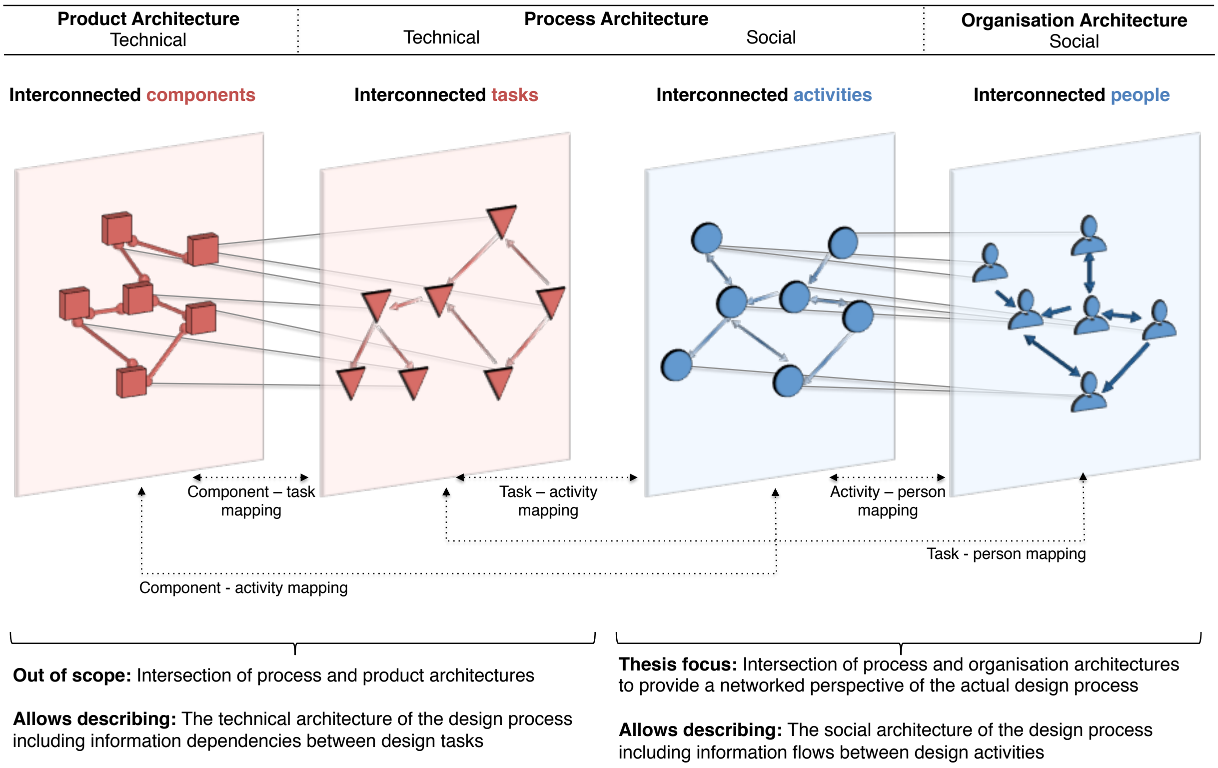


Figure 4-1: The socio-technical design process as a result of the intersection of product and process architectures, plus process and organisation architectures

4.1.1 The actual process architecture and its relationship with the planned process

At the most general level, the model I propose in this thesis provides means to describe the actual design process at the intersection of process and organisation architectures and to compare it with the planned design process at the intersection of process and product architectures (or other process architecture representations). In addition, the model connects the actual process architecture with the observed behaviours and system functions, particularly in terms of process performance measures. (See figure 4-2).

As illustrated in figure 4-2, the actual process architecture is divided between its actual composition and actual structure. As previously argued, this combination of composition and structure is an important driver of actual behaviour, which is observable in what has been termed 'patterns of designing' (Clarkson and Eckert, 2005, p. 18). The expressed behaviour not only generates the desired function (the design outputs), but also influences the actual process architecture, dynamically modifying its structure and/or composition (e.g. Davidsen 1992; Gero & Kannengiesser 2007; Sterman 1994). Combined with impacts from the environment, the feedback from actual behaviour to actual architecture is the source of the actual process architecture's evolution and subsequent changes in actual behaviour (Norman and Kuras, 2006), and is one source of the 'partially evolved' nature of the design process of engineering systems (de Weck et al., 2011, chap. 6). Although this model considers composition and structure as distinct aspects of the architecture, it recognises that a system's composition can dynamically affect its structure, and vice versa.

The planned process architecture, composed of a planned composition and a planned structure, can be derived from detailed models such as the Signposting Framework, more conventional task network models such as process DSMs, or tacit knowledge obtained through the focal firm's experience with previous projects. The planned process architecture also provides necessary information to design the formal process, including original lists of activities (or tasks), project members, and managerial roles and responsibilities. This planned process architecture, which constitutes the starting point of the actual process architecture, is based on a desired function and assumptions about the type of architecture that can deliver behaviours that match the intended functions.

Feedback mechanisms between the planned and actual process architectures allow for dynamic adjustments to the actual process and the intended plans. The more clearly the firm and the participants can receive this feedback, the better will be the dynamic adjustments and the decision-making process (Busby, 1998).

The interactions between actual composition and actual structure and the feedback from behaviour to architecture—generating the evolved process architecture—occur dynamically after the project begins; however, their precise description or explanation is beyond the scope of this thesis. To simplify the model, I focus here on the strongest paths for these influences, as shown by the numbers and directional arrows in figure 4-2. Thus, the overall sequence can be read as follows:

1. A planned design process architecture is defined, allocating resources and generating an original guide for the design process.
2. The actual process architecture emerges, with the process architecture plans and the assigned resources as key inputs. Because every detail of the process architecture can never be fully captured, and because the environment also shapes the architecture, the original planned architecture will tend to diverge from the actual process.
3. The actual process architecture will generate a range of behaviours, some with a function and a contribution to the design process objectives.

4. The design process participants are able to observe part of the actual behaviour and the actual function produced by such behaviour. Decision makers will contrast the actual function against the desired function to assess whether the process is delivering what is required and is performing satisfactorily.

5. If decision makers or others who can affect the design process detect a negative misalignment between plans and reality, they can modify the planned process architecture to improve the actual function. This modification may involve reallocating resources and redesigning the process, with the hope of driving the process closer to its desired function. In practical terms, a desired function would be equivalent to a design output that is on time, on budget, and on specifications, and therefore, equated with process performance.

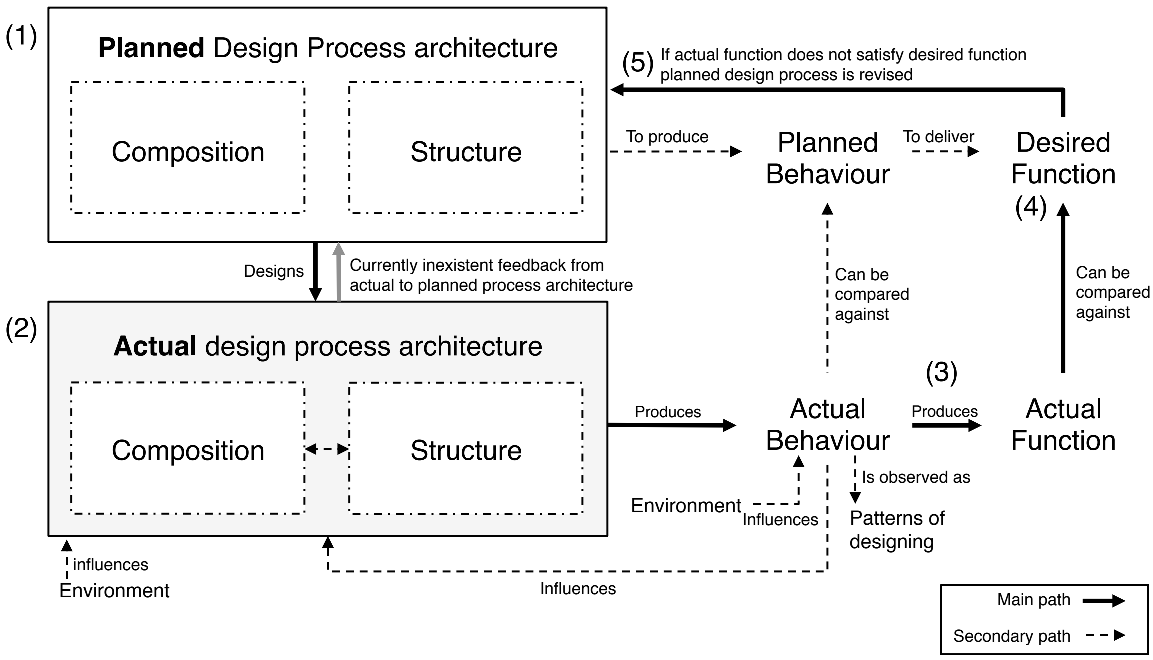


Figure 4-2: The actual design process architecture and its dynamic relationship with the planned process architecture

Based on this relationship between actual and planned process architectures, a more efficient feedback loop of the two processes is possible. For example, the actual process architecture could be directly modelled and quantified so that at each point in the process the actual could be compared with the planned process. Although previous approaches have not provided sufficient detail to make this feedback possible, a dynamic, multilevel characterisation of the actual design process architecture can be achieved through systematically gathering design patterns produced by the actual process behaviour.

4.1.2 Modelling the intersection between process and organisation architectures

With this overall model as a starting point, two important questions remain unanswered: **What exactly is the actual process architecture?** and **How can we derive the actual process architecture?** The proposed approach, summarised in figure 4-3, models the actual process architecture as a combination of the actual organisation architecture and the actual 'process-organisation architecture', which is the intersection of process and organisation architectures.

The structure of the actual organisation is based on information-driven interactions between project participants. These interactions are information exchanges required either to perform an activity or resolve the information dependency between two activities. The composition of the actual organisation is acquired from the attributes of each of the listed project participants. For example, the process may include 10 people, some from different departments and/or organisations, and may include a particular proportion participants with different roles, hierarchies, professional and academic backgrounds, seniorities, and so on. This compositional makeup, which cannot be captured simply through the network structure, is relevant because it impacts the development process (e.g. Sosa 2014; Chen & Gable 2013; Reagans & Zuckerman 2001).

In turn, for the architecture mapping process to organisation, the structure is based on the actual affiliations of project participants to activities (the mapping of people to activities). This structure allows for many-to-many relationships, or in other words, one activity may be performed by many people and one person may perform many activities. The mapping of people to activities may include not only affiliations but also the intensity the affiliation in terms of formal responsibility for the activity, time spent, or frequency of work in the activity. The composition of the actual process-organisation architecture includes the list of activities and their attributes (usually first defined through a model of the planned process architecture) and the list of people (which should match the composition of the actual organisation architecture).

All these elements, which form the actual composition and structure of the process architecture (information exchanges, mapping of people to activities, compositional diversity, etc.), are captured through an examination of the patterns of designing that are directly observable in the actual process behaviour.

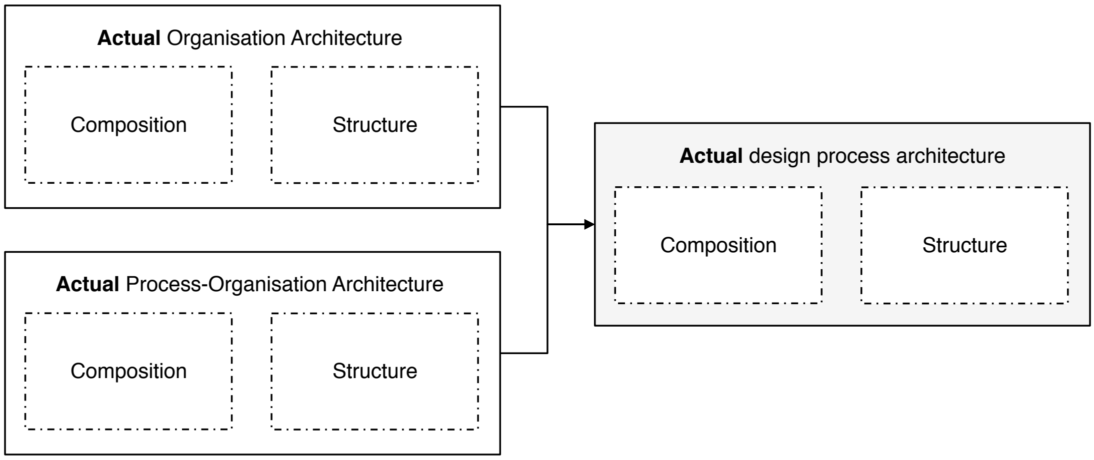


Figure 4-3: Inputs required to model the actual design process architecture

4.1.3 Activities in the context of the actual process architecture model

To clarify how activities are integrated into the model and to elaborate on the working definitions from the literature review, we can expand on the dynamic aspects of the relationship between people and activities. For example, the working definition in 2.3.1 states that an activity comprises a set of actions, executed individually or in a team to transform a set of information inputs into a set of information outputs. But how does the model treat an activity over time? Do all people involved in an activity always participate together? Does an activity appear only once in the project timeline, or can it be distributed and reappear in different periods of time?

Based on actual patterns from the two case studies, and considering the possibilities and limitations of the network-based models and methods introduced in 2.2, this model treats activities as follows:

- An activity is considered active as long as somebody is executing actions that constituent part of that activity.
- Although an activity may include a total of, say, 10 people during the whole engineering design process, these 10 people may never work concurrently together on that activity. In one time frame, there may be two people concurrently involved, and in another one, three, or five, and so on.
- An activity can freely switch from active to inactive during the engineering design process. An inactive period is one in which nobody executes activity actions. The model does not assume that this fluctuation necessarily corresponds to iterations. An activity's recurrence

does not necessarily involve rework, and such a recurrence also could be a required and planned step towards the activity's completion.

- The model's macro scale allows for people to be simultaneously involved in more than one activity. For periods of a day or a week, a person is perfectly able to be simultaneously involved in different activities; however, if the time periods are much shorter and the detail higher, a person may be able to execute only one activity at a time.

Figure 4-4 presents a graphical summary of these temporal dynamics, using a fictional process with five people (P) and four activities (A).

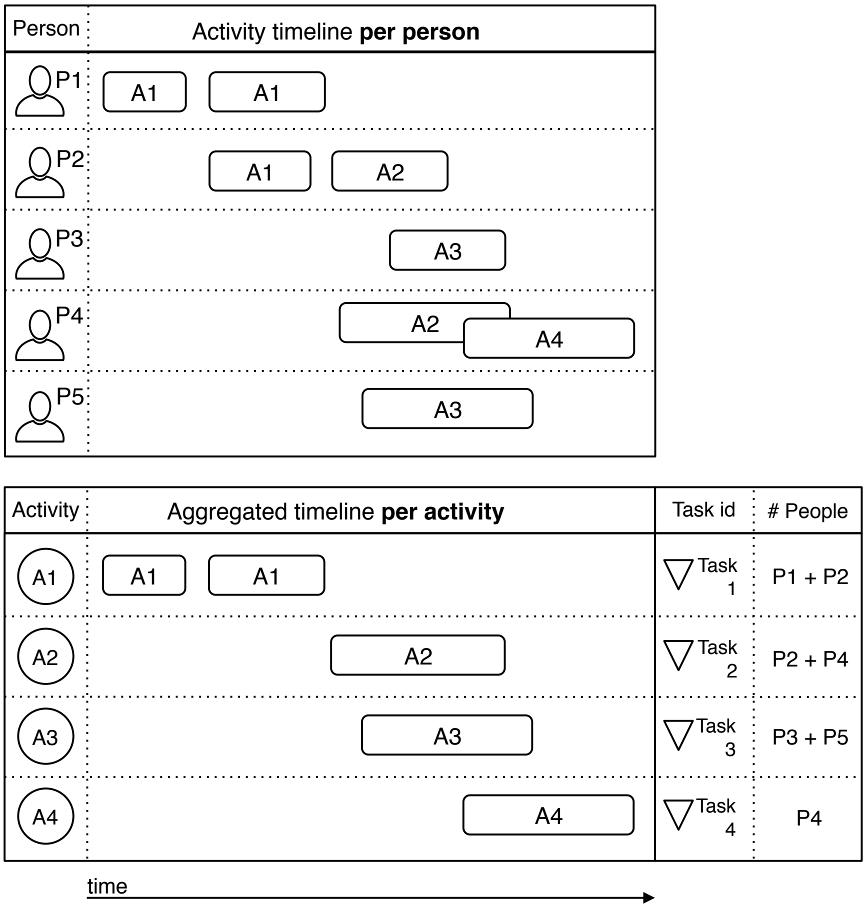


Figure 4-4: Illustration of per person and aggregated activity dynamics in the context of the actual process architecture model

4.1.4 Deriving information flows between activities

The combination of the actual organisation architecture and the actual process-organisation architecture provides the actual process architecture, with activities indirectly connected through project participants. Figure 4-5 shows a static example using a simplified diagram of a simple process composed of three activities (A_1, A_2, and A_3) and three people (P_1, P_2, and P_3). In this example, only P_2 and P_3 directly exchange information. P_1 works on A_1 and A_2, P_2 works directly only on activity A_2, and P_3 works directly only on activity A_3.

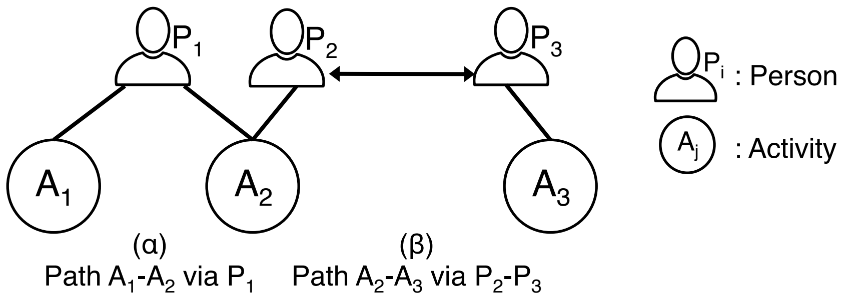


Figure 4-5: Simplified diagram for the model of the actual process architecture

Although this approach does not reveal the information dependencies between activities, we can infer the actual process architecture, and therefore, the actual information flows, based on the multiple paths available to exchange information between activities. The diagram labels the two available paths for information to flow between activities as α and β. The first path, α, corresponds to the direct flow of information between activities, A_1 and A_2 via P_1, who participates in both activities. The second path, β, occurs when project members P_2 and P_3, who participate in different activities, A_2 and A_3, exchange information about those activities.

Using this model, particularly the paths for actual information flow between activities, we can reproduce the actual process architecture in a way that aggregates all information flow paths into a representation showing only activities. Such a representation is important because only through a representation that directly connects activities can we compare the actual and the planned process architectures.

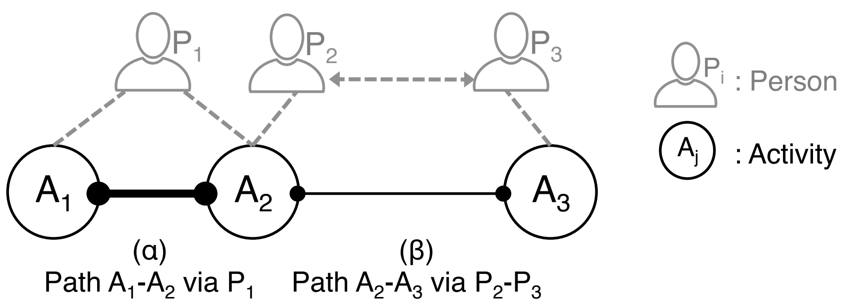


Figure 4-6: Simplified diagram for the model of the actual process architecture. Information flows between activities are derived based on information exchanges between people and their affiliations to design activities

Figure 4-6 shows a projection of the actual process architecture, connecting activities directly based on their inferred information flows. Based on the actual information exchanges and activity-people affiliations, there is a direct information flow between A_1 and A_2 represented byα, and a direct information flow between A_2 and A_3 represented by β. Considering that information between A_1 and A_2 flows directly through only one person, α is modelled as stronger than β, where information flows only indirectly through exchanges between P_2 and P_3. For this example, the model indicates no actual information flow between A_1 and A_3. In reality, of course, information might still flow between A_1 and A_3 not through direct information exchanges between individuals, but through documents or other systems that store and exchange information. In that case, the model can recognise this acquisition of information as an asynchronous information exchange between individuals. For example, if P_1 produces a document that P_3 uses as an input for A_3, then an information flow between A_1 and A_3 can be recognised and weighted according to its influence.

This rather simple example covers only the most basic aspects of the actual process architecture model. To include the remaining aspects, we must consider additional information obtained through the structure and composition of the actual organisation architecture and the actual process-organisation architecture. This additional information includes:

From the actual organisation structure → The strength of the information driven-interactions between the project participants: The information flow between two activities changes depending on the intensity or strength of people's connections across activities. Also, the more people who bridge a pair of activities, the more actual information can be exchanged. To incorporate this aspect, the model modifies the weight of each information flow between activities accordingly.

From the actual process-organisation structure → The strength of the affiliation between the project participants and the activities they perform: People who spend more time and have higher levels of responsibility on a particular activity exercise more influence on the activity and its information flows. Therefore, the model modifies the weight of the information flows between activities accordingly.

From the actual organisation composition → The attributes characterising each process participant: The information flow between activities can be modified according to the heterogeneity of those participating in the exchange. Although the weight of information flows does not need to be modified based on participants' diversity, the model accounts for this compositional diversity as an attribute that can affect information flow characteristics.

The time dimension: Both the actual organisation architecture and the mapping of process-organisation can be modelled as dynamic networks, and therefore, the actual information flow between activities can be dynamic and its evolution can be explored. The proposed approach accounts for these temporal dynamics in consistency with the dynamic activity aspects previously introduced.

4.2 Analytical methods to quantify the actual design process architecture

At the most general level, the analytical method proposed here provides a way to systematically quantify the actual process architecture model. The goal is to enable data-driven comparisons with the planned process architecture and to test relationships between the actual process architecture and the actual process performance.

The key metrics used to characterise the process architecture structure include network size, network density, group betweenness centralisation, and node betweenness centrality. To quantify and weigh in relative terms all possible paths between activities, I selected Stephenson and Zelen's (1989) information centrality and its associated centrality nearness matrix. For the compositional characterisation, the metric I selected to quantify compositional diversity was Agresti and Agresti's (1978) index of qualitative variation (IQV). These metrics and the rationale for their selection were introduced and discussed in the literature review, section 2.2.3.

Figure 4-6 provides a simplified example of the required inputs to implement the method, using process architecture DSMs, organisation architecture DSMs, and process-organisational DMMs (implicit in the mapping of process to organisation). The information utilised is represented by the red paths and the people and activities to which the paths are connected. The stacked matrices represent the temporal dimension of the analysis; each layer contains information for one time period. In practice, the dynamic network analysis would be implemented through a graph-based approach; therefore, the matrices in figure 4-7 are only illustrative.

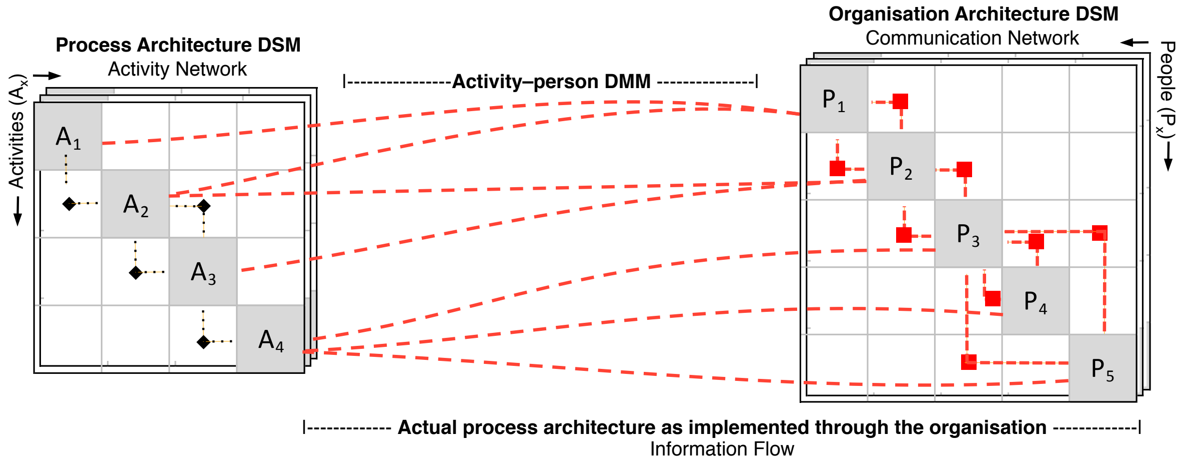


Figure 4-7: The actual process architecture presented as chronologically stacked DSM matrices. The figure includes information exchanges (organisation architecture), the mapping of people to activities (activity-people DMM), and the list of activities (process architecture)

The method provides a compositional and structural characterisation for activities and interfaces utilising cross-sectional data. In contrast, longitudinal data is used to characterise the structure and composition of the whole design process architecture. The rationale for this difference is that to analyse and interpret the characterisations of each level of analysis two important things are required: 1) meaningful variability in the independent variables (the architecture) and 2) a dependent variable and/or benchmark that matches the independent variable.

At the activity level, the variability of the independent variable is obtained through changes in the structure and composition of the organisation network performing the activity, such as differences in the number of people (size), connectivity between people (density), and diversity of people (index of qualitative variation) across activities. The dependent variable is the activity's reported performance, measured through aspects such as meeting deadlines, staying on budget, and matching specifications.

At the interface level, the variability of the independent variable is obtained through changes in the structure and composition of the organisation network performing the interface (which are the same as those at the activity level). The dependent variable can be measured based on whether interface problems exist.

At the whole process level, the variability of the independent variable cannot be obtained by comparing multiple processes because the examined design process occurs only once, is expected to be unique, and cannot be directly compared with any other process. As a result, the variability of the independent variable is obtained through examination of the evolving structure and composition of the process architecture as a whole, recording metrics of the independent variables over time. Because a static dependent variable has no use in this case, the evolving process architecture can be benchmarked only against dynamic dependent variables. Such dynamic dependent variables include models of the planned process architecture, project milestones, and/or prescriptive process models describing process stages and their respective network configurations. Figure 4-8 summarises the levels of analysis alongside their independent variables (network architecture) and dependent variables (performance or comparison base).

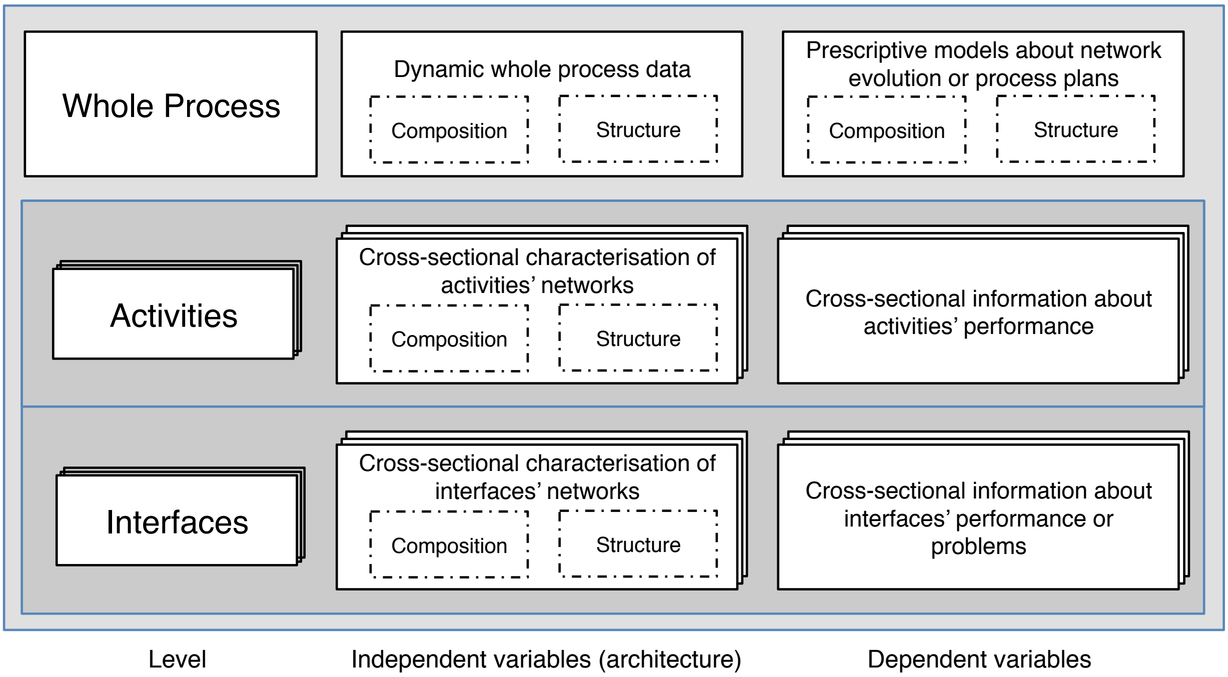


Figure 4-8: Levels of analysis and respective independent and dependent variables per level

One framework three levels of analysis

Next, I detail the framework for the three levels of analysis: activities, interfaces, and whole design process network. For each level, specific aspects of the conceptual model and the analytic method are covered. Concluding this chapter, section 4.5 integrates the perspectives gained from the three levels of analysis and the comparison between actual and planned process architectures.

4.3 Networked activities

Overview

As described in section 2.3, a number of network-based process models have been developed to guide the design of engineering systems. Many of those models, such as the one shown in figure 4-9, represent design activities only as single nodes, treating activities as black boxes that receive information inputs and deliver information outputs. As a result, the architecture that actually delivers the information transformation—the network structure and the activities' composition—remains invisible and inaccessible for further analysis and actionable insights. However, if we were to unfold the architecture of activities, we would be able to see the organisation network through which people conduct activities. The architectures of such organisation networks matter because they affect behaviour and ultimately the activities' performance outcomes. Phrased differently, a well-crafted activity architecture should contribute to more desirable design outputs and performance; therefore, it is essential to characterise these architectures, and through these characterisations, to learn which architectures seem to be the most suitable for each activity.

With this background, we revisit the activity level sub-questions posed in section 3.2.2:

- How can we **model** the actual architecture of **activities**? (RSQ 1a)
- How can we **quantitatively characterise** the actual architecture of activities? (RSQ 2a)
- How can we test the relationship between the architecture of activities and their **performance**? (RSQ 3a)

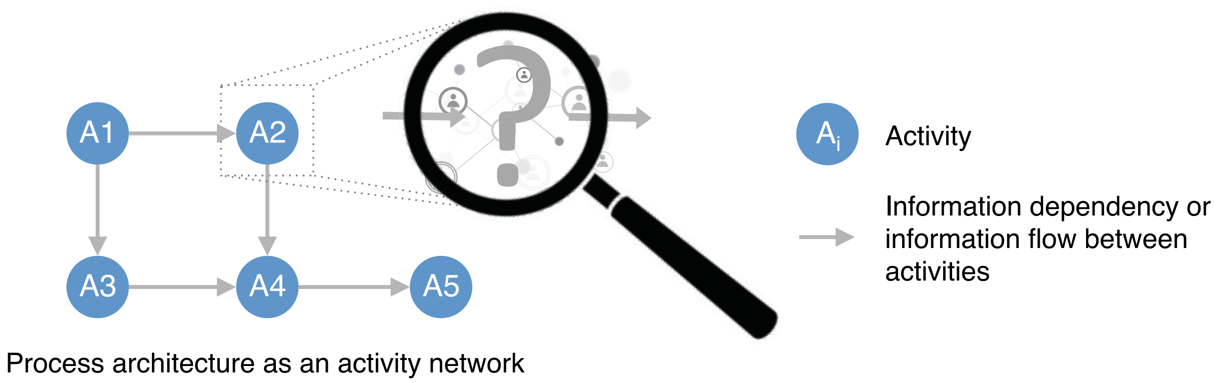


Figure 4-9: The invisible network architecture of each design activity

4.3.1 The model: design activities as organisation networks

A missing piece in the design process literature is the middle ground between models that provide a detailed characterisation of each activity and those that characterise the whole process architecture. On one hand, detailed activity characterisations such as those in ethnographic studies (e.g. Bucciarelli 1988) or in experimental laboratory studies (e.g. Cardella et al. 2006) often are not scalable, given their resource-intensive nature. On the other hand, whole process architecture models are scalable but do not provide a network characterisation for each activity. This disconnection between process architecture and architecture focused on activities limits our understanding of how the organisation of each activity affects performance outcomes and complicates the identification of best practices and potential problems at each activity level.

To unfold the architecture of each activity and bridge detailed activity characterisations with overall process architecture approaches, the model I propose uses the overall organisation architecture and the mapping of people to activities revealed through observable patterns of designing to build each activity's network architecture (figure 4-10).

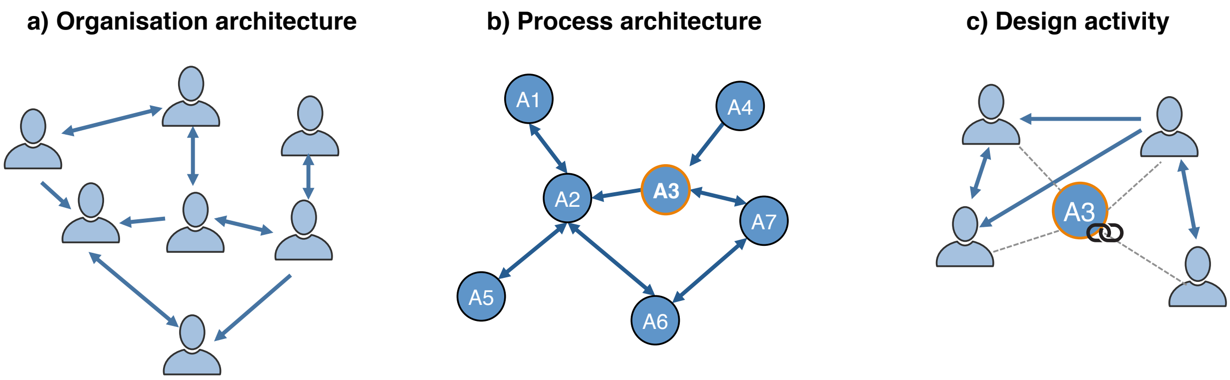


Figure 4-10: Conceptual model of a design activity: a) shows an organisation network based on information exchanges; b) shows an activity network based on information dependencies between engineering design activities; c) shows a design activity in which people's affiliations with A3 define the composition of the activity participants, and the information exchanges between people (from the organisation architecture) define the structure of the design activity network.

Based on the overall conceptual model for the actual process architecture introduced in 4.1 and the relationships between architecture and performance reviewed in 2.2.6, I selected the following aspects as the main elements to characterise an activity:

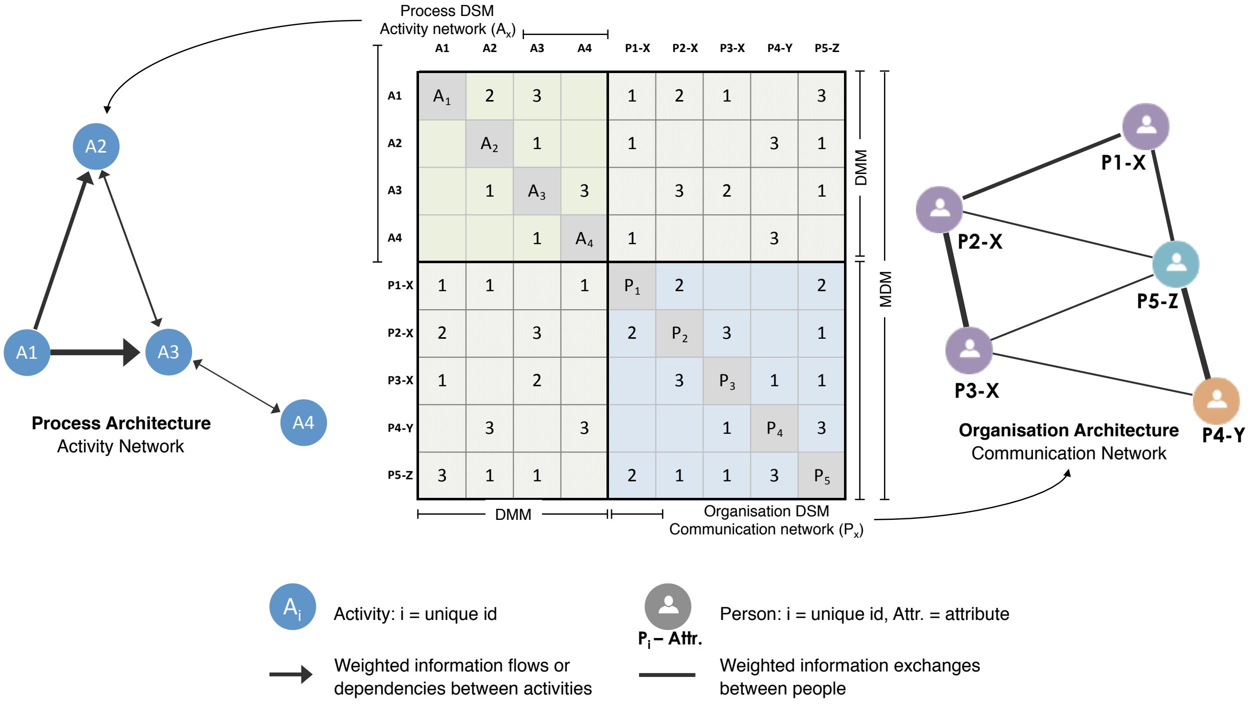
- The number of people participating in each activity, because the size of an organisation can impact aspects such as coordination and availability of knowledge (e.g. Chen & Gable 2013; Tsai 2001)

- The connectedness of the people involved in each activity, because the organisation network's density can affect the efficiency of information flows (e.g. Burt 1992; Easley & Kleinberg 2010)
- The compositional diversity of people, because heterogeneous groups can provide a more diverse knowledge pool but also may pose communication challenges (e.g. Jansen et al. 2006; Rodan & Galunic 2004)
- The degree of intensity and actual workload distribution and each participant's responsibilities in an activity, because these factors can reveal patterns about how activities are organised and can affect the overall density of connections within the activity's architecture

Figure 4-11 shows a fictitious example including the information inputs required to execute the model. These inputs are:

- The organisation architecture (bottom left quadrant of the Multi-Domain Matrix (MDM) in figure 4-11) as an undirected and weighted organisation network, plus selected attributes of the process participants, captured from questionnaires, email metadata, and/or other communication tools
- A weighted mapping of activities to people in the form of a Domain-Mapping Matrix (DMM) based on participants' affiliation to activities and the strength of their affiliations, captured from questionnaires and/or process mining of activity logs
- Optionally, the process architecture (top left quadrant of the MDM) as a weighted and directed activity network, obtained through interview or process mining, used to contextualise the place each activity occupies in the overall process architecture, although not required to characterise each activity's actual architecture

With these inputs we can **build a model for the network architecture of each activity** that captures aspects of the structure and composition of each activity that are often unaccounted for in other models.



	A1	A2	A3	A4	P1-X	P2-X	P3-X	P4-Y	P5-Z
A1	A_1	2	3		1	2	1		3
A2		A_2	1		1			3	1
A3		1	A_3	3		3	2		1
A4			1	A_4	1			3	
P1-X	1	1		1	P_1	2			2
P2-X	2		3		2	P_2	3		1
P3-X	1		2			3	P_3	1	1
P4-Y		3		3			1	P_4	3
P5-Z	3	1	1		2	1	1	3	P_5

Figure 4-11: A fictitious example illustrating the weighted information inputs to build the model. All matrices follow the inputs in columns convention (IC)

Figure 4-12 shows the results of applying the model to the example presented in figure 4-11. More specifically, figure 4-12 illustrates how activities 1 and 2 (A1, A2) can be **unfolded to reveal their inner architecture**. For instance, if we take attributes X, Y, and Z as departments and the weighted affiliation as a combination of responsibility and amount of hours that each person spends on the activity, A1 is characterised as a network of four people (P), three from department X and one from department Z. On a scale from one to three, P1 and P3 are affiliated to the activity with strength of one, P2 with a strength of two, and P3 with a strength of three. Only P1 and P3 have no direct work-related information exchanges, while all the other members exchange information directly with a variety of intensities.

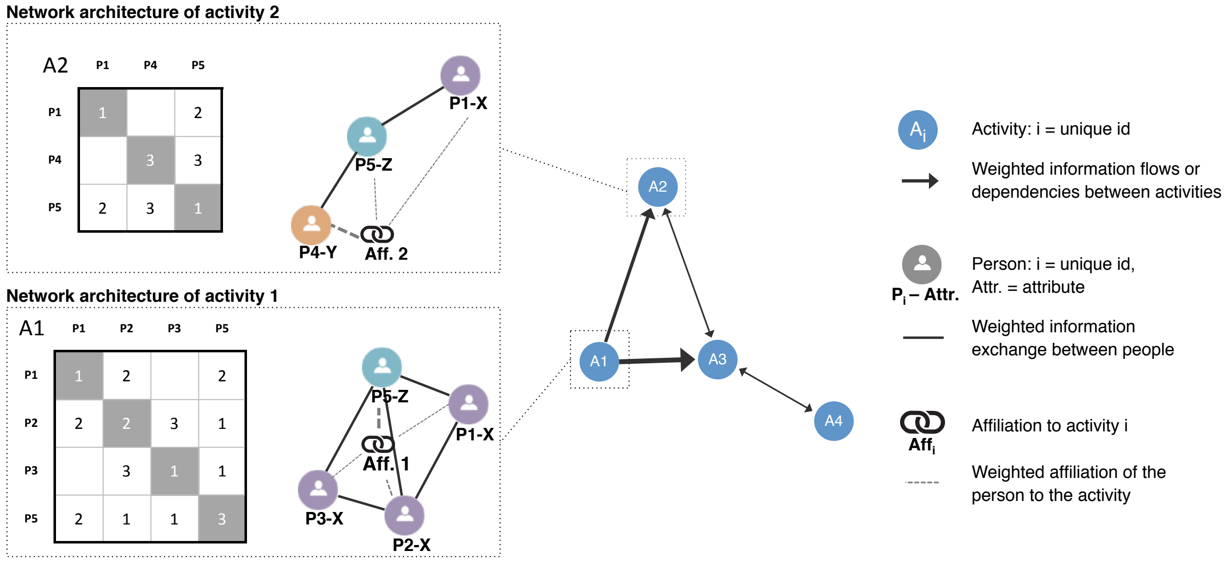


Figure 4-12: Results of applying the model to unfold activities A1 and A2, using data from example in figure 4-11. On the left, a modified weighted organisation DSM and a network graph show the network architectures of activities A1 and A2. The diagonal of the matrices show a measure for the affiliation strength of each individual to the activity.

4.3.2 The method: quantitative characterisation and analysis of activities' architectures

To advance to a systematic characterisation and an analysis of the activity architectures, we need a method to quantify these architectures based on structure and composition. Once the activity architectures are quantified, we need another method to analyse patterns across these architectures, enabling the assessment of the relationship of those patterns with performance measures at the activity level.

Activity characterisation

To quantify **structure**, I use the following measures: **network size** as the number of people participating in the activity, and **weighted network density** as the sum of all edges in the activity's network architecture, divided by the theoretical maximum of the sum when such network is fully connected (Wasserman and Faust, 1994, p. 101). To capture the degree of affiliation to the activity, I include the edge that accounts for the strength of the affiliation as part of the network when calculating density (diagonal of the matrices in figure 4-12).

To quantify **composition**, the framework uses the **Index of Qualitative Variation (IQV)** (Agresti and Agresti, 1977), which provides a measure of network heterogeneity from zero to one. Zero indicates no heterogeneity (all participants have the same attribute), and one indicates the maximum heterogeneity (each participant has a different attribute).

Table 4-1 provides a quantitative characterisation for the architectures of A1 and A2, based on the data provided in figures 4-11 and 4-12 and the quantitative methods.

Table 4-1: Quantitative characterisation of the actual architecture of fictional activities A1 and A2

		Activity 1	Activity 2
Structure	Size	4	3
	Density	0.52	0.56
Composition	IQV	0.56	1

While size, density, and IQV capture primary aspects of the architecture of activities and are simple to interpret, a number of other network and non-network metrics also can be used to quantitatively characterise the model proposed. These alternative metrics include those discussed in section 2.2.4 and others identified in reviews of complexity metrics in engineering design, such as Kreimeyer and Lindemann (2011).

Analysis of activity architectures

Once all activities are characterised in terms of size, density, and IQV, we can examine the activity architectures for emergent patterns. Subsequently, we also can test for a relationship between a detected pattern and the associated performance measures at the activity level. To start, we can cluster activities in groups according to the quantified characteristics (size, density, and IQV for compositional diversity). If quantitative performance measures of efficiency and/or effectiveness are available for the activities, these measures can be calculated per cluster and used to test for statistically significant differences among the clusters. If differences indeed exist, they can be used as evidence that certain types of activity architectures are related to certain performance outcomes.

One method to find clusters in a set of activities is the two-step cluster analysis. Using this method, each activity is assigned to one cluster based on the compositional and structural characteristics of its organisation network in relationship to the characterisations of all the other activities. A one-way ANOVA test can be used afterwards to test for significant performance differences among the resulting clusters. If the clusters in the one-way ANOVA test show a statistically significant difference in performance, then a more detailed analysis can follow to clarify potential causality between architecture and performance. This analysis, in turn, provides the basis for targeted interventions to the activities' architectures. Even if statistically significant differences do not exist, the clusters can still be used to generate groups of activities that share similar features, and therefore, are likely to require different management strategies.

An alternative to analyse the effect of an activity architecture on performance is through a regression analysis. This method uses the architectural features of the activities as independent variables and available measures of performance as dependent variables. The advantage of this method is that it allows the modelling of a function for the relationship between the variables, and if there is a statistically significant relationship, the obtained function can be used as a predictor for the relationship between the variables. The disadvantage is that regression analysis on its own is not suitable for the identification of groups (clusters) of architectural features that affect performance in a non-linear form. Due to the expected interaction effects between the independent variables and the expected non-linearity of their effects on performance (observed in related studies), the combination of clustering and one-way analysis of variance to test for statistical differences among the clusters provides a more suitable approach that can be complemented by regression-type analyses to gain additional insights.

Figure 4-13 summarises the proposed steps to implement the methodological approach to activity characterisation and analysis.

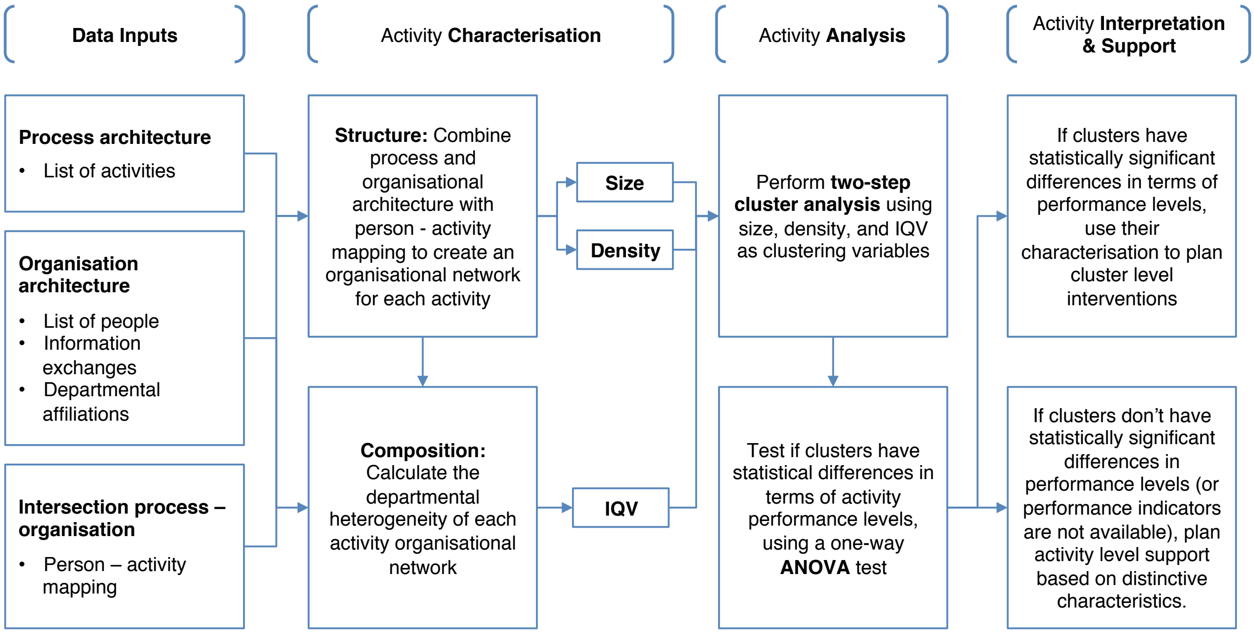


Figure 4-13: Summary of the proposed approach to characterise activities. The conceptual model generates a compositional and structural characterisation of activity architectures. The analytical method quantifies the characterisation and provides the basis for interpretation and decision-making support.

4.4 Networked interfaces

Overview

The engineering design process literature provides guidance on how to identify, map, and analyse design activities and their information dependencies; however, a systematic characterisation of interfaces between engineering design activities is missing and the impact of an interface's characteristics on performance is unclear. To fill these gaps, I propose a new approach to characterise process interfaces as networks of interactions between members of interfacing activities. In addition, I provide guidance on how to test and interpret the effect of those characteristics on interface problems. As a result, I will show how the structural and compositional characteristics of the organisation network between information-dependent activities can provide valuable insights to support complex engineering design processes. I apply the proposed model and methods to the descriptive case study to reveal a relationship between the structural and compositional characteristics of the process interfaces and reported interface problems. Implications of this approach include the integration of information about process and organisation architectures, the possibility of systematically distinguishing key network characteristics associated with interface problems, and improved support to interface managers through a better overview of the actual information flows between activities.

My assumption is that an improved approach to interfaces will enable a detailed examination of the mechanisms underlying system integration during the design process, the identification of relationships between interface characteristics and interface problems, and improved support to the project management of engineering design processes.

With this background, I revisit the interface level sub-questions posed in section 3.2.2:

- How can we **model** the actual architecture of **interfaces** between activities? (RSQ 1i)
- How can we **quantitatively characterise** the actual architecture of interfaces? (RSQ 2i)
- How can we test the relationship between the architecture of interfaces and their **performance**? (RSQ 3i)

A new perspective on process interfaces

The approach to process interfaces takes into account that engineering design is a social process of information transformation, comprised of a set of interdependent design activities and people working on those activities (Bucciarelli, 1984; Hubka et al., 1988; Simon, 1996). In this context, a ***process interface*** is what allows information dependencies between design activities to be resolved. Process interfaces combine elements from the process and organisation domains: The process architecture defines when an interface between two activities is required; the organisation architecture describes how people interact to fulfil activities; the mapping of people (organisation) to activities (process) connects these two domains and makes explicit how activities interface through people to exchange and transform information. Figure 4-14 shows the relationship between these domains through a simplified representation of a process interface.

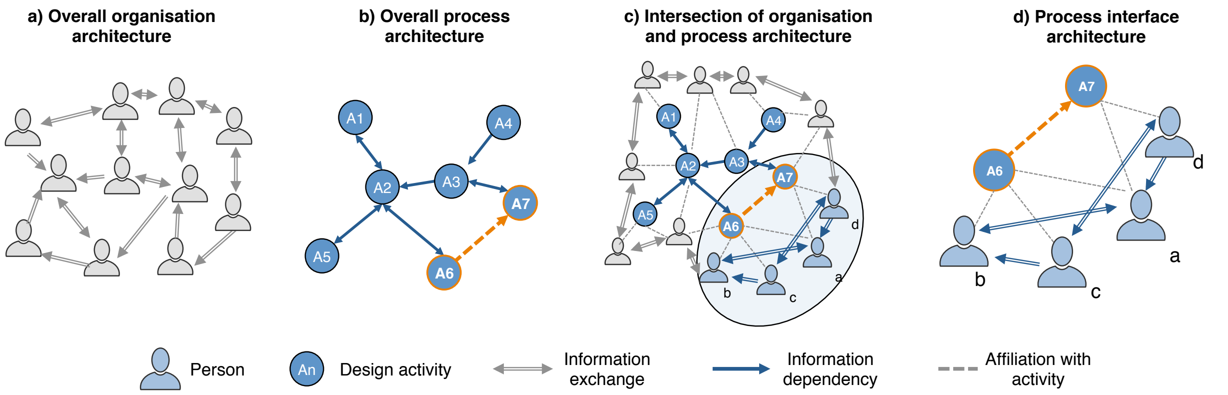


Figure 4-14: Conceptual model of a process interface: a) shows the overall organisation network based on information exchanges; b) shows the activity network based on information dependencies between engineering design activities; c) shows the intersection between the organisation and process architectures; d) shows the process interface as a combination of a), b), and c)

The dependent variables of performance are concentrated in process interface problems resulting from inadequate information exchanges and/or inadequate information transformation processes between information-dependent activities. Such problems have been associated with product integration difficulties that are likely to lead to significant negative impacts on time, budgets, and quality (Browning and Eppinger, 2002).

Three things are essential to examine process interfaces: 1) a model to conceptualise and characterise process interfaces, 2) the means to quantitatively analyse the interface characteristics, and 3) a basis to interpret and provide data-driven support based on the quantitative analysis. The next section discusses these three requirements.

4.4.1 The model: process interfaces as organisation networks

Based on the characteristics of process interfaces, I propose a model in which **each process interface is described as an organisation network of information exchanges between two information-dependent design activities**.

As shown in figure 4-14, and inspired by studies that combined elements from the process and organisation domains (Christian, 1995; Morelli et al., 1995; Sosa, 2008; Sosa et al., 2007b), the proposed model describes each process interface as a bimodal network of two activities and people. These networks are constituted by two information-dependent activities, a number of people interconnected via information exchanges, and the mapping of each person to either one or two activities at each side of the interface. To facilitate analysis and interpretation, activities are grouped into cohesive work-packages, based on the subsystems being designed. Subsequently, each activity group is associated to one of three macro categories previously introduced: integrative work activities, integrative subsystem activities, and modular subsystem activities. With this information and the application of standard network analysis metrics such as density and size, we can characterise the structure of each process interface network (Borgatti et al., 2013; Wasserman and Faust, 1994), and in turn, with attributes such as each person's functional group affiliation and the type of activity on each side of the interface we can characterise their composition.

Figure 4-15 shows an application of this model to a fictional interface between activities A_1 and A_2. The key inputs for the characterisation are:

- Activity A_2 requires information from activity A_1.
- Five people (P_x) are involved in this interface, and their information exchange interactions are described in the organisation interaction matrix (ii).
- P_1 and P_2 are affiliated directly only with A_1. P_3 and P_4 are affiliated directly with both activities. P_5 is affiliated only with A_2. This information is obtained from the person-activity affiliation matrix (iii).
- P_2, P_4, and P_5 are from the functional area 'engineering', P_1 from 'quality assurance', and P_3 from 'project management'.

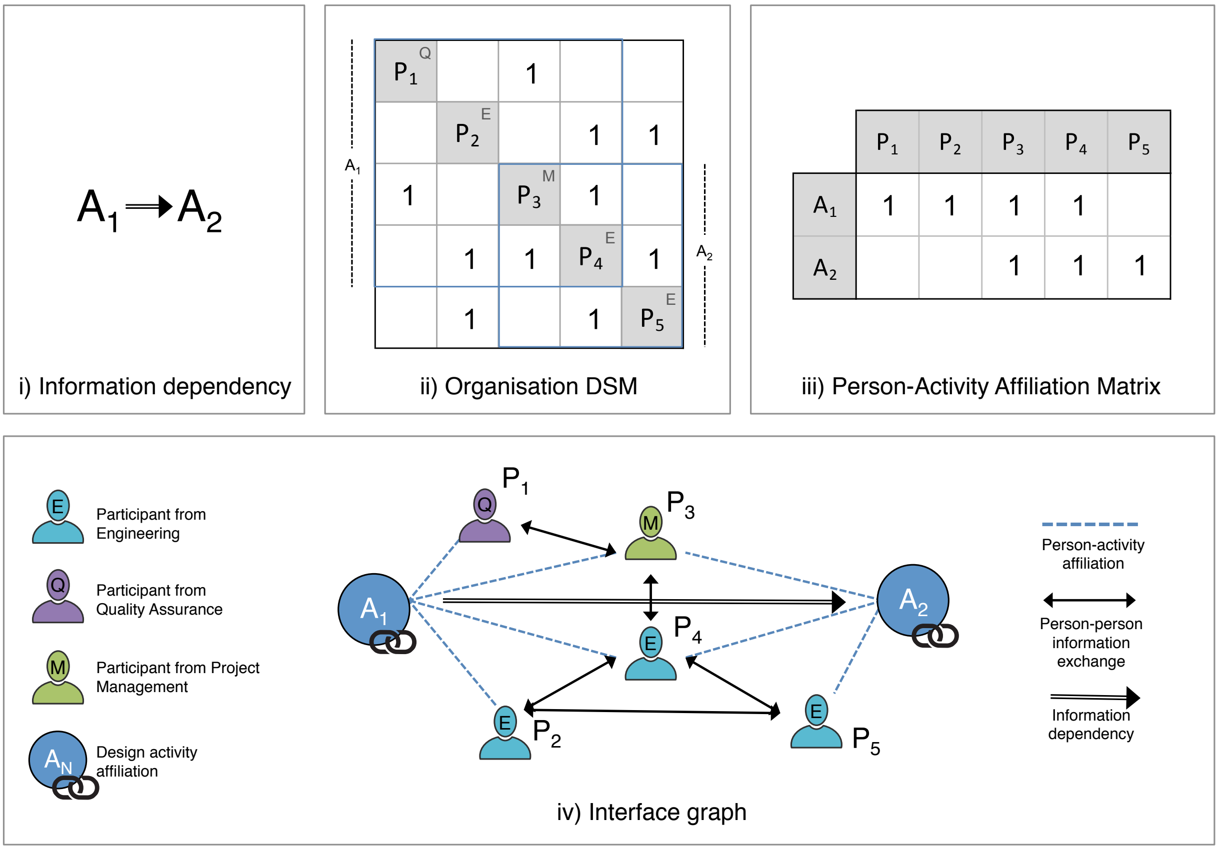


Figure 4-15: Information inputs for the analysis of process interfaces (generic example): i) shows the direction of the information dependency between activities A1 and A2; ii) shows the organisation DSM matrix for people P1 to P5; iii) shows the person-activity affiliation matrix; iv) shows the process interface graph generated as a result of combining i), ii), and iii)

Using the previous inputs, table 4-2 shows a simplified quantitative characterisation of the interface $A_1 \rightarrow A_2$, including the full set of structural and compositional aspects used to characterise each process interface.

Table 4-2: Compositional and structural characterisation of a process interface, based on the characteristics of interface process network in figure 4-15. The value for compositional diversity is calculated using the Index of Qualitative Variation (IQV)

	Structural			Compositional
	Size	**Ties**	**Density (Size/Ties)**	**Compositional diversity**
Explanation	Number of people	Number of reported interactions between people	Ties divided by number of possible ties: 5/10	1 participant from quality assurance 3 participants from engineering 1 participant from project management
Result	5	5	0.5	0.84 (see IQV formula in 2.2.3)

When analysed in the context of all project process interfaces, the simple set of metrics in table 4-2 can be used to describe each interface and as an input to compare all project interfaces and identify potential problems.

4.4.2 The method: from interface characterisation to analysis and interpretation

To advance from an overall process interface characterisation to a systematic analysis and interpretation, we need a method to quantify compositional diversity and analyse interfaces in the context of the whole design process in which they are embedded. Here, the same methods utilised for activity characterisation can be followed, including a combination of two-step clustering and one-way ANOVA to systematically analyse process interfaces based on their structural and compositional characteristics. The same consideration about performing a regression analysis as introduced in subsection 4.3.2 (networked activities) also applies here.

Figure 4-16 shows a graphical summary for the set of proposed steps to implement the methodological approach to process interface characterisation and analysis.

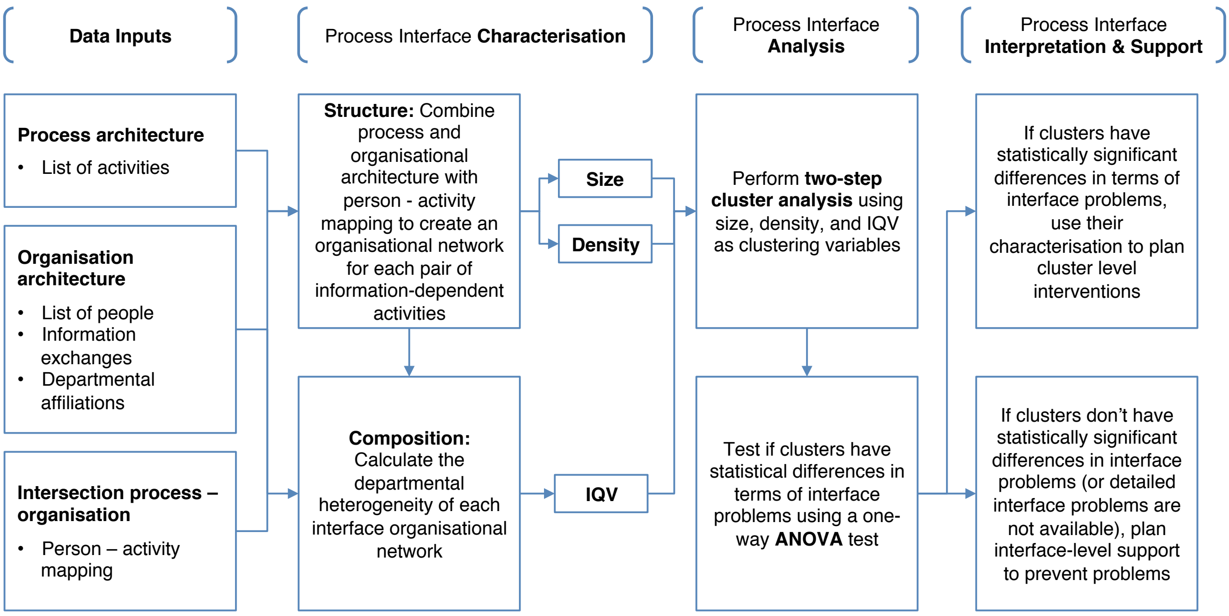


Figure 4-16: Summary of the approach to analyse process interface analysis. The conceptual model allows to generate a compositional and structural characterisation of process interface architectures. The analytical method allows to quantify the characterisation and provides the basis for interpretation and decision making support

4.5 Networked process

This level of analysis explores the way in which information flows through a network of interdependent design activities. The analysis characterises the aggregated and temporal dynamics of information flows between activities as they are implemented through the network of people in the project. A bimodal dynamic network model and a set of quantitative methods were developed to quantify information flow among activities and identify patterns at the whole project level. As a result, the model summaries all structural aspects captured at the activity and interface levels.

With this background, we can review sub-questions posed at the whole process level in section 3.2.2:

How can we **model** the actual architecture of the **whole design process**? (RSQ 1w)

How can we **quantitatively characterise** the actual architecture of the **whole design process**? (RSQ 2w)

What is the relationship between the **dynamic architecture** of the **whole design process** and its **planned design stages**? (RSQ 3w-2)

How can we quantitatively **compare** the **actual** and the **planned** architecture of the **whole design process**? (RSQ 3w-1)

At the whole process level, the model used to compare the actual and planned process architectures is generally the same as that used to examine the relationship between the dynamic architecture and the design stages, although the emphasis and methods in each case differ. Therefore, I begin this section with the model and methods developed to compare the actual and the planned architectures of the whole design process (4.5.1), and then move to the model and methods used to examine the relationship between the dynamic architecture and its planned design stages (4.5.2).

4.5.1 Comparing the actual and the planned design process

Based on the architecture of the process domain introduced in section 2.4, both the social and the technical angles of process architecture can be described in terms of what is planned and what is actual. However, the most difficult but also useful comparison is between a planned process architecture, defined as a task-based network driven by technical information dependencies, and an actual architecture, defined as an activity-based network driven by social information flows. Such a comparison is difficult because it requires bridging two different angles of the design process architecture, the social and the technical, and therefore, involves comparing models with different data and assumptions. But despite the additional difficulties, this comparison is particularly useful because with it we can identify whether the relative intensity of the information dependencies and the intensity of the information flows between any two tasks are consistent. For these reasons, my focus here is on comparisons between planned task networks and actual activity networks; however, the same principles can be applied to compare the actual design process architecture against any other activity or task-based network model of the design process.

Other noticeable comparisons between actual and planned architectures include research in what has been termed 'socio-technical congruence' (Cataldo et al., 2008), multi-domain alignment measures such as the 'coordination deficit' proposed by Gokpinar et al. (2010), and the identification of matched, unattended and unidentified interfaces introduced by Sosa et al. (2004). While these comparisons provide means to quantify actual and planned architectures, none of them fills all the requirements and gaps identified in this thesis. Requirements that include enabling a multilevel comparison between the actual and the planned design process architecture (not between different domains) that is not constrained by a one-to-one mapping between the organisation and the process domains.

A prerequisite for comparing the actual activity network against the planned task network is for both networks to be defined at the same level of detail: Each task must have an equivalent activity in which the specified work is performed. The task network describing the planned process architecture (based on technical information dependencies) can be modelled using any of the network-based approaches introduced in the literature review, including those directly mapping tasks that use process Design Structure Matrices (DSMs) (Eppinger and Browning, 2012; Steward, 1981), or the more detailed approaches that explicitly consider parameters between tasks, such as the Signposting Framework (Clarkson and Hamilton, 2000; Wynn et al., 2006). The key input required from the planned process architecture is a representation of the information dependencies between tasks that can be mapped in matrix form. In contrast, it is not sufficient to model an activity network describing actual process architecture with conventional approaches that directly connect activities, such as process DSM models or graph-based activity networks. The reason for this lies in the limitations of current models of the actual process architecture (described in 1.1.3 and 2.7). In those approaches the information flows between each pair of activities are modelled only as single edges, based on estimations and/or process plans, making them inappropriate as models of the actual process. Consequently, I have chosen instead to use the model and methods previously introduced in this thesis to map the actual process architecture.

Once the model of the planned process architecture (based on a task network) and the model of the actual process architecture (based on an activity network) are defined, they must then be compared.

A method to compare actual and planned process architectures

The proposed method to compare the actual and planned process architecture has five steps:

1. Obtain the task network that represents the planned process architecture.

   This task network can be directly acquired through interviews with expert members of the engineering design project. In such interviews information dependencies between planned tasks can be structured through a process DSM, workflow diagrams, or any other network-based model. The task network may be valued and directed. If it is valued, it should be normalised on a scale from 0 to 1. Information dependencies between tasks, if valued, should be weighted to reflect their strength, not their criticality (to keep them comparable to valued information flows).

2. Map the information of the actual process architecture obtained at the intersection of process and organisation.

   Utilising the model introduced in this chapter, the full network of information exchanges between project participants is combined with the cross-domain affiliation network that maps people and activities. Both networks can be valued; however, both should be undirected in order to compute the next step; that is, they should be symmetric or symmetrised .

3. Identify and quantify in relative terms all paths in the actual process architecture.

Stephenson and Zelen's (1989) information centrality algorithm can be used to compute the information centrality nearness matrix of the actual process architecture obtained in step 2. This matrix provides a weighted representation of all possible paths for information to flow between every point in the graph. The more paths, and the smaller the degree of separation between nodes, the higher will be the information flow's assigned value. This algorithm is defined only for symmetric matrices, which is why the actual process architecture must be symmetrised in step 2.

4. Normalise the values of indirect paths between activities to obtain a relative measure for information flows.

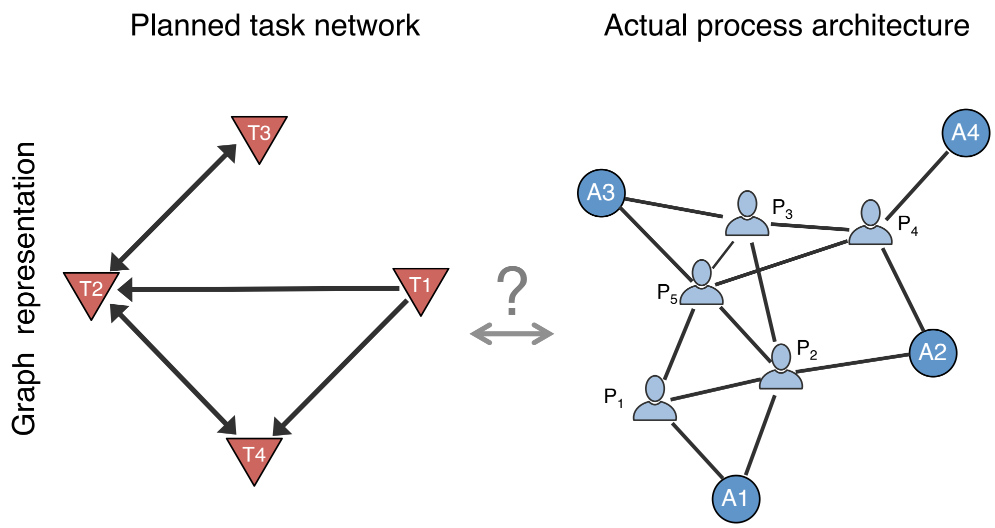
   Now that information flows have been calculated between each node in the graph, we can obtain the information flows between each pair of activities and normalise their values on a scale from 0 to 1. This normalised, undirected, and value square matrix represents the actual information flows between activities and includes all available information about the actual information exchanges between project participants and their degree of participation in each activity.

5. Calculate the difference between the normalised matrix of information flows (activity network) and the normalised matrix of information dependencies (task network).

   The normalised matrix of actual information flows minus the normalised matrix of planned information dependencies between tasks yields a new matrix with a quantitative indication of the relative alignment between the actual and planned process architectures. Each non-diagonal cell in this new matrix indicates in relative terms the result of information flow minus information dependency. The lowest values point out lower-than-expected information flows compared with the intensity of information dependencies. The highest values indicate that some of the information-driven interactions across activities can be redirected to strengthen the lower relative information flows.

Simplified application:

Let us consider a design process with four tasks (T_i) and their planned information dependencies, four matching activities (A_j), five people (P_k) and their information exchanges, and cross-domain affiliations between people and activities (see figure 4-17). We can employ step 1 and 2 above to obtain the planned task network and the actual process architecture. Without additional work, we cannot directly compare the planned and actual process architectures, because the planned task network shows only directly connected tasks and the actual process shows indirectly connected people and activities (that is, people intermediate the flow of information between activities).



Matrix representation

	T1	T2	T3	T4
T1	T1	1		1
T2		T2	1	1
T3		1	T3	
T4		1		T4

	A1	A2	A3	A4	P1	P2	P3	P4	P5
A1	A_1				1	1			
A2		A_2				1		1	
A3			A_3				1		1
A4				A_4				1	
P1	1				P_1	1			1
P2	1	1			1	P_2	1		1
P3			1			1	P_3	1	1
P4		1		1			1	P_4	1
P5			1		1	1	1	1	P_5

Figure 4-17: Set of inputs to compare planned and actual process architecture. The figure depicts the graph-based and matrix-based representations of a fictional example. At this stage, no quantification is yet available about the actual information flows between activities

To quantify the relative strength of the information flows (implementing step 3), the actual process architecture presented in figure 4-17 must be transformed to an information centrality nearness matrix in which the values indicate the relative closeness between the nodes (see figure 4-18). For example, considering all possible paths and their weights (which for simplicity are all binary), the points farthest apart are A1 and A4, a distance that can be corroborated graphically looking at the graph. In contrast, the strongest connection in the graph is between P1 and A1, because P1's attention is divided among fewer elements than the others.

Also, A1 involves only two people, which strengthens the A1-P1 connection in relative terms (closely followed by A4-P4).

Planned task network
Matrix representation

	T1	T2	T3	T4
T1	T1	1		1
T2		T2	1	1
T3		1	T3	
T4		1		T4

Actual process architecture
Information centrality nearness matrix

	A1	A2	A3	A4	P1	P2	P3	P4	P5
A1	A1	-0.08	-0.12	-0.25	0.169	0.085	-0.08	-0.14	-0.04
A2	-0.08	A2	-0.11	-0.06	-0.08	0.02	-0.05	0.05	-0.06
A3	-0.12	-0.11	A3	-0.17	-0.07	-0.05	0.093	-0.06	0.066
A4	-0.25	-0.06	-0.17	A4	-0.22	-0.17	-0.1	0.163	-0.12
P1	0.169	-0.08	-0.07	-0.22	P1	0.067	-0.05	-0.11	0.021
P2	0.085	0.02	-0.05	-0.17	0.067	P2	5E-04	-0.06	0.012
P3	-0.08	-0.05	0.093	-0.1	-0.05	5E-04	P3	0.008	0.039
P4	-0.14	0.05	-0.06	0.163	-0.11	-0.06	0.008	P4	-0.01
P5	-0.04	-0.06	0.066	-0.12	0.021	0.012	0.039	-0.01	P5

Figure 4-18: Application of the Stephenson and Zelen information centrality nearness matrix to calculate weighted information flow paths. Colour scale represents relative closeness, with green the highest closeness and red the lowest

Figure 4-19 illustrates the final two steps. Step 4 is represented in the normalised actual activity network, and step 5 is the subtraction of actual from planned. Here, the lack of a direct information flow between A1 an A4 is important because the planned task network clearly identified an information dependency between T1 and T4, a dependency that does not appear to be appropriately addressed. In addition to this diagnosis, other insights can facilitate the alignment between the planned and the actual. For example, when a relative surplus of information flow is identified, such as between A3 and A4, resources may be reconfigured to address an information dependency, such as between T1 and T4. A simple option may be to create a role for P4 in A1, or if that is inconvenient, to foster information exchanges between P4 and P2, in which P2 could provide information about A1 and P4 about A4.

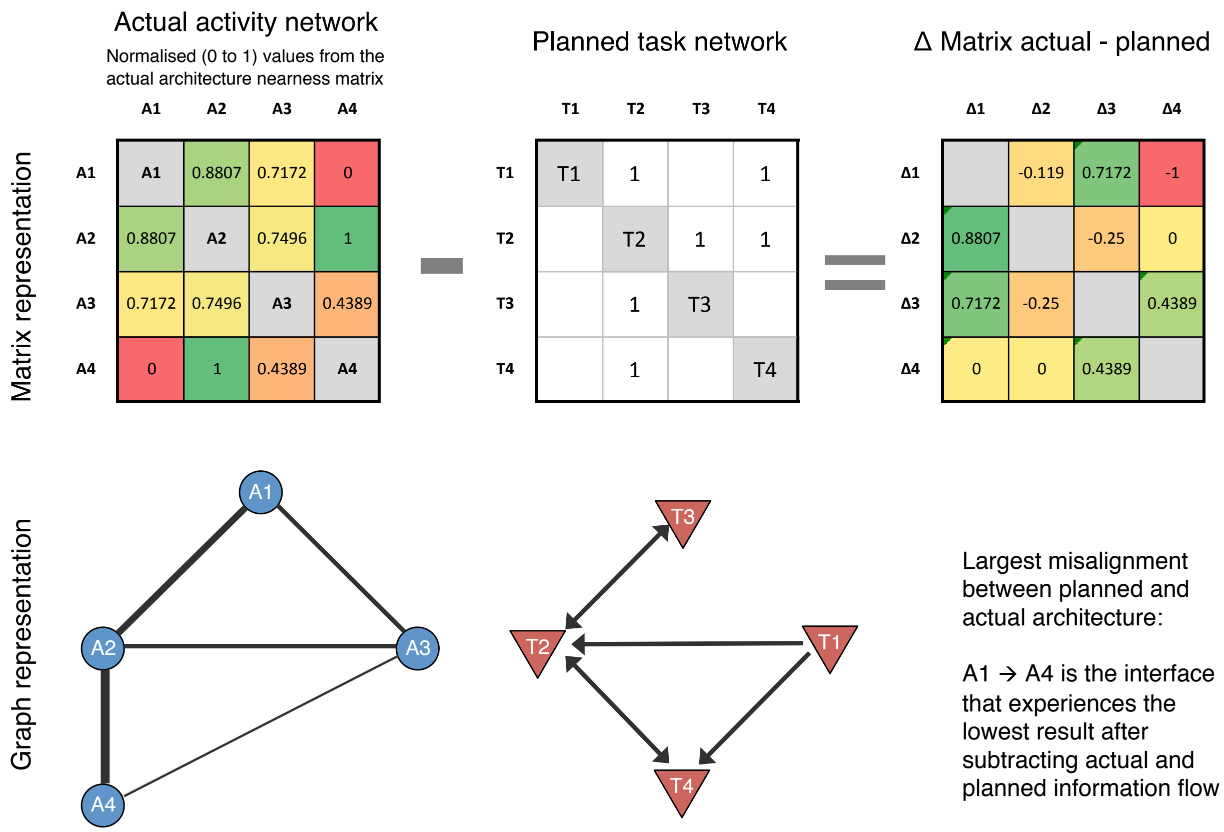


Figure 4-19: Actual normalised activity network, planned task network, and the subtraction of actual - planned. Although in the planned process, T4 requires information from T1, the actual process indicates no direct information flow between A1 and A4. Colour scale represents relative closeness.

4.5.2 The dynamic architecture of the whole design process

Previous research on design process architecture has not considered the evolution of information flows through the various stages of an engineering design process, nor has it clearly distinguished between the architectures of the planned and the actual information flows between activities. To fill this gap, the NPr Framework aims to (1) develop a conceptual model and a way to quantify the dynamic architecture of the whole process design process, and to (2) propose a generic architecture per engineering design stage against which the actual architecture can be interpreted.

A premise is that the patterns of information exchanges between people and the subsequent information flows between activities change over time as the project proceeds through various stages (Eppinger and Salminen, 2001) (for a graphical example, see figure 4-20). However, without the means to quantify, analyse, and interpret these patterns, we cannot assess how these patterns change, and whether they follow predictable changes between stages, are affected by stage transitions, can be used to assess deviations from what should be expected, or exhibit other meaningful associations with process performance.

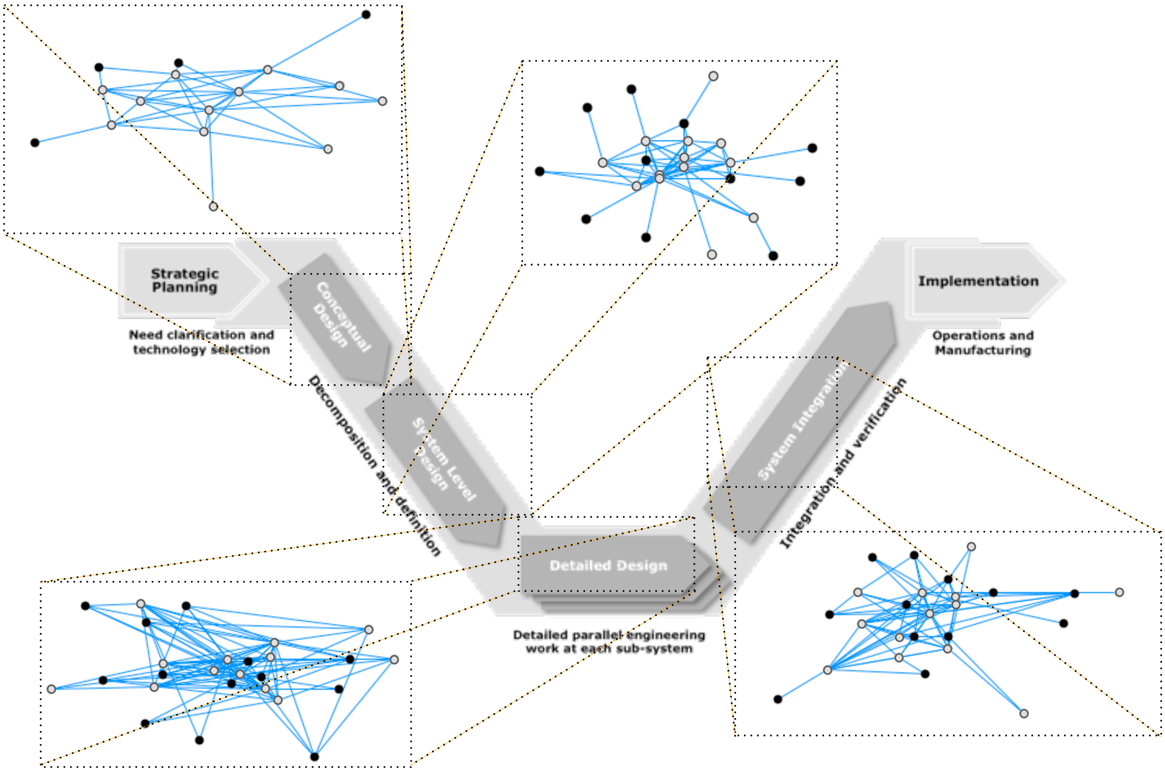


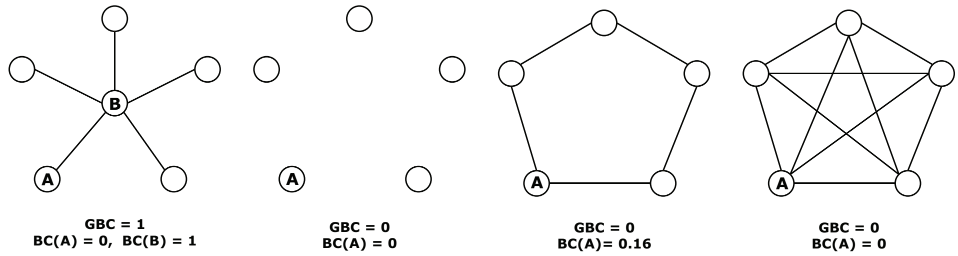
Figure 4-20: The evolving actual process architectures through the stages of an engineering design process, on the background simplified SE V model. Each network graph provides a snapshot of the process architecture per process stage.

The evolving information centrality of each activity is measured to quantify information flow patterns between activities. In turn, the evolving information centralisation of the whole network of activities is measured to quantify the evolution of the whole process architecture. Following the networked framework at the intersection of the process and organisation architectures, the actual process architecture is calculated by combining information from the process and the organisation architectures for each time period, which could be daily or weekly, depending on the resolution of available data. Although the actual computation applies graph-based network analysis, the procedure can be more easily illustrated as in figure 4-7, where there is one organisation DSM and one process-organisation DMM per time period.

Modelling the **information centrality of each activity** is important because it reveals which activities intermediate information and when, and therefore, which ones are more likely to influence the information transformation process. In turn, changes in this information centrality shape the temporal dynamics of the design process and affect the development of critical interfaces between subsystems (Braha and Bar-Yam, 2004). An activity's information centrality depends on its degree of intermediation in information exchanges, which the framework quantifies using the network metric, node Betweenness Centrality (BC) (Freeman, 1977).

Modelling the **information centralisation of the overall design process** reveals how the distribution of activities' information centrality evolves during the process, and thus, indicates how vertical or horizontal the distribution of information centrality through the design process is. Such distribution of information indicates the actual degree of process modularity, based on information exchanges that act as a powerful summary measure calculated at each time period. This information centralisation of the whole process is quantified using the network metric known as Group Betweenness Centralisation (GBC) (Wasserman and Faust, 1994, pp. 189–192). Table 4-3 summarises these two measures.

Table 4-3: Summary of the concepts of information centrality and information centralisation

	Description of Network Measure	Meaning for Information Networks
Activity information centrality through node Betweenness Centrality (BC)	Proportion of shortest paths from all nodes to all others passing through the node in question. If all paths have to go through the node, the value is 1; if there is always an alternative path, the value is 0.	Activities with high betweenness centrality are more likely to act as intermediaries in information exchanges, and therefore, exercise more control or influence on those exchanges.
Whole process information centralisation through the Group Betweenness Centralisation (GBC)	Distribution of betweenness centrality across the nodes. The index reaches its maximum value (1) for the star graph where the entire network has one central point. Its minimum value (0) occurs when all nodes have exactly the same betweenness centrality.	High group betweenness centralisation is a sign of a centralised information exchange architecture, in which only one or a few groups of activities intermediate most information exchanges. Low group betweenness centralisation is an indication of decentralised, horizontal information flows.
Graphical Summary		

Although a more traditional process DSM approach can be used to obtain the actual process architecture, employing experts' direct knowledge regarding how activities are implemented (Browning, 2002; Eppinger and Browning, 2012), the inter-temporal nature of this analysis would make such a task overly difficult for those asked to provide the required information. The problem originates in the multiple ways in which activities can be implemented and connected to other activities through people. Instead of directly asking experts to provide the dynamic network of task interactions, I propose utilising a bottom-up approach, first acquiring the mapping of people to activities over time, then the interactions between people, and finally composing a unified network structure—all information that often can be obtained from digital data traces or that does not require expert judgement.

Part of the complexity of large engineering design projects results from the multiple intertwined processes executed in parallel. To facilitate interpretation of the results, the process architecture can be simplified by combining low-level activities into larger activity packages and categories based on their common work in developing a particular subsystem or a sub-process. For this purpose, I use the hierarchical breakdown of the process domain, introduced in section 2.3.1 (figure 2-9), which can describe the design process dynamically at multiple levels of analysis. However, the empirical results presented in the framework application will focus only on activity categories and the whole network of activities for illustrative and practical purposes. Nonetheless, with the same model and data, we can analyse information flows at a higher level of detail, such as single activities, or map how information flows through the organisation domain (people, teams, and departments), integrating the effect of the process architecture.

The relationship between design process stages and the dynamic network structure of information flows

The model presented provides a way to empirically quantify the changing patterns of information centrality between activities and the overall information centralisation in engineering design projects. However, to interpret the model's empirical results, we must have a base against which to compare the obtained information control and centralisation patterns.

One option is to compare the empirical results against a previous, closely related project to which the same quantification of information flows was applied. Although this option provides a direct benchmark, it does not yield a theoretical understanding of information flow patterns. In addition, data from closely related, successful projects is often unavailable. An alternative option is to build a comparison from qualitative descriptions found in generic models of system engineering stages. As long as the engineering design project under study follows some sort of stages, such as those described by the SE-V model, we can benchmark against idealised information control and centralisation patterns extracted from each generic stage. To enable a comparison between empirical results produced by the framework and system engineering stage models, we must translate the qualitative system engineering stage descriptions and characteristics into expected information flow patterns. For this purpose, I developed two assumptions:

A) A relationship exists between system engineering stages and the whole information network topologies.

Each stage's information network topology is defined by the stage's degree of system decomposition or integration, the amount of activities, and the dominant activity categories expected at the stage.

Stages with low levels of decomposition or high levels of integration tend to have a more centralised information network topology. Conversely, stages with high decomposition or low integration tend to have more decentralised information network topologies. This relationship is consistent with the notion that integration requires centralised coordination, while decomposition decentralises the information network to allow for parallel work at the subsystem level (Haskins et al., 2011; Hossain, 2009).

The greater the number of activities involved, the lower the overall centralisation of the information network, which is consistent with an empirical relationship observed between the number of elements of a network and its density and centralisation (Anderson et al., 1999)

Stages dominated by integrative work activities are more likely to have higher levels of centralisation because these activities relate to project coordination (Hossain, 2009). Stages dominated by the design of integrative subsystems are more likely to have medium levels of centralisation because these activities relate to the technical integration and coordination of two or more modular subsystems (Sosa et al., 2003). Finally, stages dominated by the design of modular subsystems dominates are more likely to have low levels of centralisation because these design activities tend to be more technically specialised, and therefore, more distributed (Sosa et al., 2003).

B) A relationship exists between system engineering stages and information control at the activity category level.

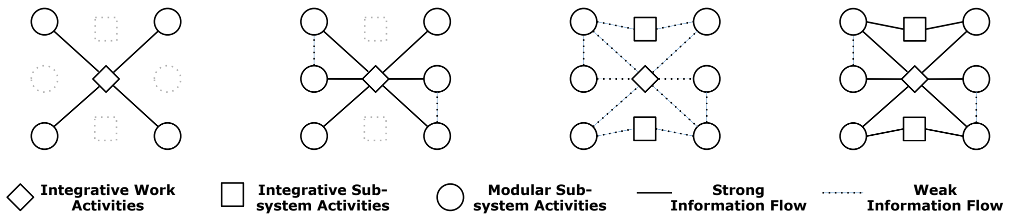
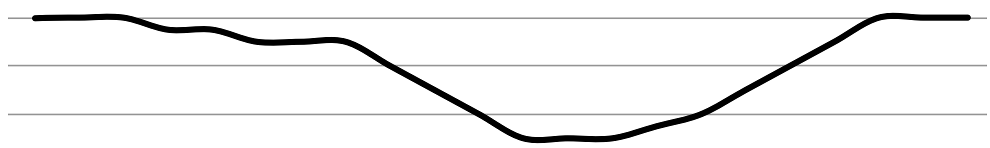
The activity categories that are more likely to centralise information flows in a given stage depend on the degree of decomposition or integration (Sosa et al., 2003) required by the stage and the stage maturity.

Low levels of decomposition or high levels of integration are primarily associated with information being centralised by integrative work activities, and to a lesser extent, with the design of integrative subsystems. High levels of decomposition or low levels of integration are associated with a relative increase in the information centralised by modular subsystem activities (Sosa et al., 2003).

The more technically detailed and mature is the work developed in a stage, the more information centrality is held by integrative subsystem activities and modular subsystem activities. In contrast, in stages where the design work is less detailed or at a higher level of abstraction, information will tend to be centralised by integrative work activities.

Building on these assumptions and utilising the system engineering stage model introduced in section 2.3.3, table 4-4 describes each of the four stages in terms of expected information centrality and centralisation patterns.

Table 4-4: Summary and comparison of expected information patterns for each stage

Information Pattern	Conceptual Design	System Level Design	Detailed Design	System Integration
Expected topology of the information network				
Activities expected to centralise information	Project management (integrative work)	Project management but with decreasing control, while modular subsystem activities increase control	Integrative and modular subsystem activities	Integrative work and integrative subsystems increase their control
Expected overall centralisation of the information network	High but decreasing; only a few areas control information	High to medium, but decreasing; information control becomes more distributed	Low and slowly increasing; information control becomes highly distributed	Medium and increasing as a few integrative areas gain information control
				

These expected patterns of information centrality and overall centralisation provide a generic, stage-by-stage base against which to compare the empirical results obtained from the application of the framework, and therefore, can be used as reference points in the application of the model and method. The application of this part of the framework, focused on the dynamics of the process, provides insights to guide design process improvements, based on identification of misalignments between information centrality and centralisation patterns at the different stages of engineering design processes.

4.6 Chapter summary

The **NPr Framework** provides means to characterise the actual design process architecture and data-driven support to the design process of engineering systems. To fulfil the identified industrial needs and knowledge gaps, the framework provides a **multilevel, dynamic characterisation** of the actual design process architecture, **enables the comparison between the actual and the planned design process architecture,** and **provides means for connecting the characterisation** of the actual design process architecture **with process performance metrics**.

To address the multilevel architecture of the actual design process, I discussed the framework according to 'networked activities' (section 4.3), 'networked interfaces' (section 4.4), and 'networked process' (section 4.5). The research sub-questions also reflect these levels of analysis. In the following, I examine the extent to which each sub-question was answered in this chapter.

Networked activities

i. **How can we model the actual architecture of activities?**

   The architecture of each activity was conceptualised as an organisation network, based on the structure and composition of each activity. This approach allowed an enriched conceptualisation of activities, which often are defined in design process models simply as a series of black boxes that receive information inputs and generate information outputs.

ii. **How can we quantitatively characterise the actual architecture of activities?** The architecture of each design activity was quantitatively characterised through the structural metrics of size and weighted density and through the compositional measure of the Index of Qualitative Variation (IQV). Because of the flexibility of the graph-based modelling approach, additional structural and compositional metrics can be easily added.

iii. **How can we test the relationship between the architecture of activities and their performance?**

   Taking into consideration the typical amount of activities and the number of independent and dependent variables that may be involved, a two-step clustering of activities, based on the architectural characterisation of each design activity, in combination with a one-way ANOVA test was proposed to examine differences in the performance measures between the clusters.

Networked interfaces

iv. **How can we model the actual architecture of interfaces between activities?**

   The architecture of each interface was conceptualised as an organisation network involving two information dependent activities. This approach can be used to model the structure and composition of each individual process interface, which often is only described as an edge in models of design process architecture.

v. **How can we quantitatively characterise the actual architecture of interfaces?**

The architecture of each process interface was quantitatively characterised through the structural metrics of size and weighted density and through the compositional measure of the index of qualitative variation (IQV). The flexibility of the graph-based modelling approach allows additional structural and compositional metrics to be added easily.

vi. **How can we test the relationship between the architecture of interfaces and their performance?**

A combination of two-step clustering of the process interfaces (based on the architectural characterisation of each interface) and a one-way ANOVA test to examine differences in the performance measures between the clusters was proposed.

Networked process

vii. **How can we model the actual architecture of the whole design process?**

The whole design process architecture was modelled as a bimodal network of activities and people that can be broken down into an information exchange network of people-people and an affiliation network of people to activities. Depending on application, this bimodal network can be dynamic or static.

viii. **How can we quantitatively characterise the actual architecture of the whole design process?**

The network metrics utilised to quantitatively characterise the architecture of the whole process model included: the betweenness centrality of activities, Stephenson and Zelen's (1989) information centrality algorithm applied to activities, the group betweenness centralisation of the whole process, and Stephenson and Zelen's (1989) information centrality matrix applied to the whole process.

ix. **How can we quantitatively compare the actual and the planned architecture of the whole design process?**

I proposed a method to use Stephenson and Zelen's (1989) information centrality algorithm and information centrality matrix to generate a process view of the actual process architecture in which activities were directly connected by information flows. This approach enabled a quantitative comparison between planned and actual process architectures.

x. **How can we test the relationship between the dynamic architecture of the whole design process and its planned design stages?**

I proposed a method to use the dynamic Betweenness Centrality and Group Betweenness Centralisation of the actual process architecture to map the dynamic evolution of an information network. Also, I developed an expected evolution of the process architecture based on standard system engineering stages to be used as a reference for evaluating the actual evolution of the process architecture.

The NPr Framework answers each of the 10 sub-questions, and therefore, is sufficiently mature to be applied to the design process of a real engineering system, so that its ease of implementation and usefulness as a source of data-driven support can be tested. This application of the framework is detailed in chapter 5.

To conclude this chapter, the framework at each of its levels of analysis is illustrated in figure 4-21. In this representation, we can see the relationship between tasks and activities, an illustration of the actual process architecture differentiating between each of the levels of analysis, and the way in which a process view of information flows between activities is derived from the bimodal representation at the whole process level (people and activities combined).

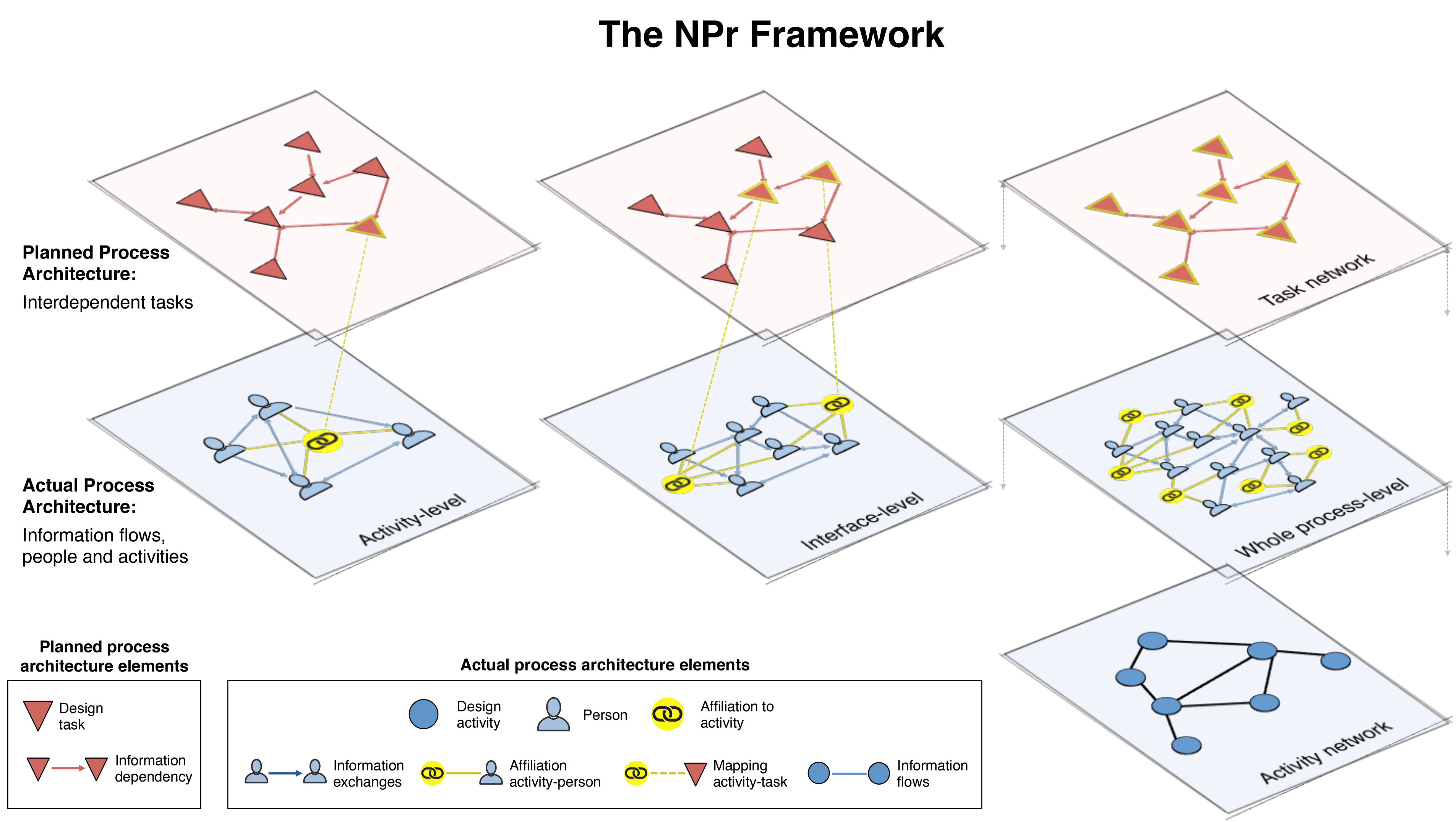


Figure 4-21: Graphical summary of planned and actual process architecture, the three levels of analysis, and how the actual activity network is derived as a result of the actual process architecture at the whole process level

5 FRAMEWORK APPLICATION

That all our knowledge begins with experience there can be no doubt
—Immanuel Kant

In this chapter I describe how the developed framework was applied to the descriptive case study to test (1) whether the framework could be applied mostly with already available information, and (2) whether the framework would yield valuable insights for the company.

The analysed design process consisted of the complete engineering design work of a biomass power plant. Although the company considered its design process as adequate and had adopted industrial standards for project and process management, the company was interested in applying this new framework because they perceived their actual process architecture as inaccessible behind the socio-technical complexities of their engineering design activities. For example, the company reported an insufficient actual process overview and general interface problems not associated with a particular domain, but located at the intersection of domains. The company believed these problems stemmed from the number of people and external organisations involved in the project, the long time spans, the multiple and complex technologies involved, and the parallel development of multiple subsystems. To provide the best overview of the case study company and to identify the sources of problems, I applied all the elements of the framework developed in chapter 4 at the levels of design activities (section 5.1), process interfaces (section 5.2), and the whole process architecture (section 5.3).

5.1 Networked activities

From the case study company's perspective, an analysis of the architecture of their design activities would allow it for the first time to test whether the characteristics of the organisation networks implementing the activities affected the performance of the activities. If an architecture-performance relationship did exist, the company could begin to identify best practices and explore the root cause of already identified performance problems.

5.1.1 Application

To apply the framework at the activity level, I used the following data sources: historical data from activity logs, questionnaire data on work-related information exchanges between core project members, data logs on activity performance including the number of document revisions per activity, and hourly budgeting systems including schedule overruns per activity.

Two performance measures served as dependent variables for each activity: scheduled hours overrun (difference between actual and planned hours) and the average number of document revisions per activity. The number of hours overrun per activity was a proxy for efficiency, and the average number of document revisions was a proxy for effectiveness. Using the method presented in 4.3.1, data was gathered to define the process and organisation architectures and to perform a mapping between them.

Process architecture data

To manage the complex engineering design process of the power plant and its many subsystems, the company assigned a unique code to each design activity. From a total of 148 unique activity codes, 44 activities were selected as suitable for a detailed analysis based on the following criteria: Two or more people were involved in each activity, and performance indicators of the activity's outcome were available.

Organisation architecture data

An electronic questionnaire was distributed to determine the communication network based on weighted, work-related information exchanges between core project members. All 49 core project participants from 15 engineering departments completed the questionnaire, yielding 756 dyads of work-related information exchanges. Consistent with the model, all reported interactions were symmetrised. The weight of each interaction was calculated based on the interaction's reported frequency and estimated impact on the design work. The engineering department to which each person was affiliated was used as a compositional attribute.

Mapping of process-organisation data

To perform the analysis, 11,742 records registering who was working when and for how long in all 44 selected activities were used to map people and activities. The bimodal network of people and activities was weighted based on the number of hours spent on the activity and subsequently normalised so that all weights for the edges were on the same scale.

Characterising the network architectures of activities

The organisation network of each of the 44 activities was characterised according to network size, density, and compositional diversity (using IQV). Table 5-1 and figure 5-1 show descriptive statistics for these variables for all selected activities.

Table 5-1: Descriptive statistics for the network characteristics of the 44 activities examined

	Minimum	Maximum	Mean	Std. Deviation
Size	2	16	7.50	4.106
Density	.00	100.00	69.7398	21.90967
IQV (0 to 1)	.000	.489	.06809	.113509

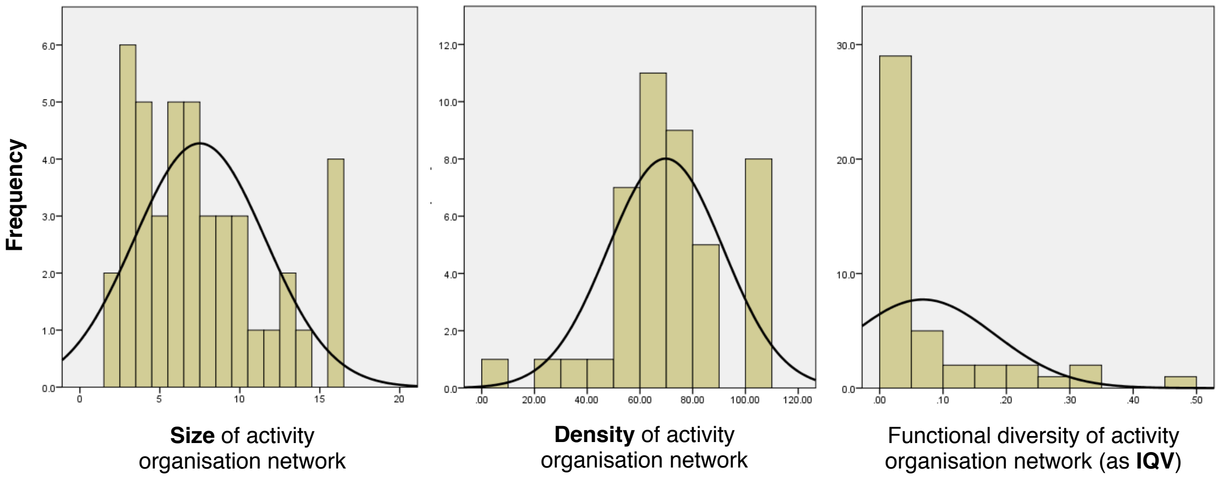


Figure 5-1: Histograms and approximate distribution curves for the network characteristics of size, density, and IQV of the 44 activities examined. Size is expressed as an absolute value of the number of participants in the design activity, density as a percentage, and IQV on its 0-to-1 scale.

To put these quantitative characterisations in context, figure 5-2 shows a graph-based illustration for three of the 44 activities. These three activities were selected to illustrate various combinations of size, density, and compositional diversity as well as the effect of weighting both the interactions between people and the strength of their affiliation with the activity. These three activities were:

- ***Activity A, flexibility calculation of structural mechanics***: This activity organisation network showed a relatively small, dense network in which all participants were from the same department. Only one dense group was distinguishable; two members (P01 and P03) steered the activity while P02 participated only incidentally.

According to performance measures, this activity had an average of 0.33 revisions per document, and compared with original estimates, used approximately 5% fewer hours than originally planned.

- ***Activity B, definition of buck stays and fixations:*** This activity organisation network was characterised by a larger, sparser network, and unlike Activity A, the members differed significantly in their network embeddedness. For example, although P05 exchanged information with all other people in the activity, P08 directly exchanged information only with P05. The activity exhibited relatively low diversity, as all members except one were in the same department.

  In performance measures, this activity had on average of 0.75 revisions per document, and compared with original estimates, used almost 50% fewer hours than planned.

- ***Activity C, evaluation of manufactured designs:*** This activity organisation network was comparatively larger and slightly more diverse, although the connectivity between its members was stronger than that of Activity B.

  In terms of performance measures, Activity C had an average of 0 revisions per document and an estimated hourly overrun of 37%.

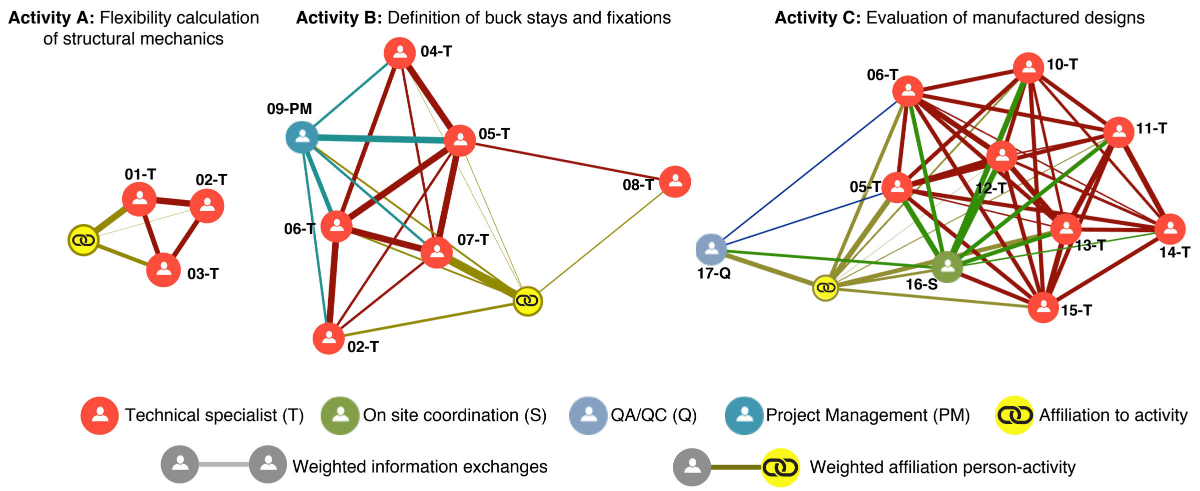


Figure 5-2: Three graphical examples of actual activity architectures. The graph layout is weighted and force-directed to represent different intensities of information exchanges. Project members and their respective edges are coloured according to departmental affiliation.

Although a one-by-one examination of each activity based on their architectural characteristics (such as the one above) can yield preliminary conclusions about the relationship between activity characteristics and performance, a more robust and scalable approach is required to systematically associate and test those relationships. The proposed method achieves such an approach with a) a two-step clustering analysis in which similar activities are grouped, followed by b) a one-way ANOVA test through which to assess the significance of changes in the selected performance indicators across the identified clusters.

Exploring the relationship between activities' architecture and their performance

a) Two-step clustering analysis

A two-step clustering analysis was performed to identify groups of activities with similar architecture configurations and to analyse if those clusters had performance differences. For the cluster analysis, the 'distance measure' used was log-likelihood, and the number of clusters was determined automatically based on the Schwarz's Bayesian information criterion (BIC). Three statistically distinct groups with a 'silhouette measure of cohesion and separation' of 0.5 were obtained, which is within the boundary between good to fair cluster quality (values between -1 and 0.2 are poor, between 0.2 and 0.5 are fair, and values between 0.5 and 1 are good [Tan 2006]). As illustrated in figure 5-3:

- Cluster 1 contained 61.4% of all activities (27) and was characterised by activities with a large number of people, a low to medium density, and low heterogeneity. The average number of document reviews and hours overrun indicated an average performance level.
- Cluster 2 contained 20.5% of all activities (9) and was characterised by activities with a small number of people, high density, and low heterogeneity. This cluster had the best performance in the process.
- Cluster 3 contained 18.2% of all activities (8) and was characterised by activities of medium to high density and medium to high heterogeneity. Activities in this cluster did not display a distinctive pattern in terms of network size. Based on the number of activities in this cluster and especially in comparison with cluster 2, this cluster concentrated a higher average number of document reviews and had a higher percentage of hours overrun.

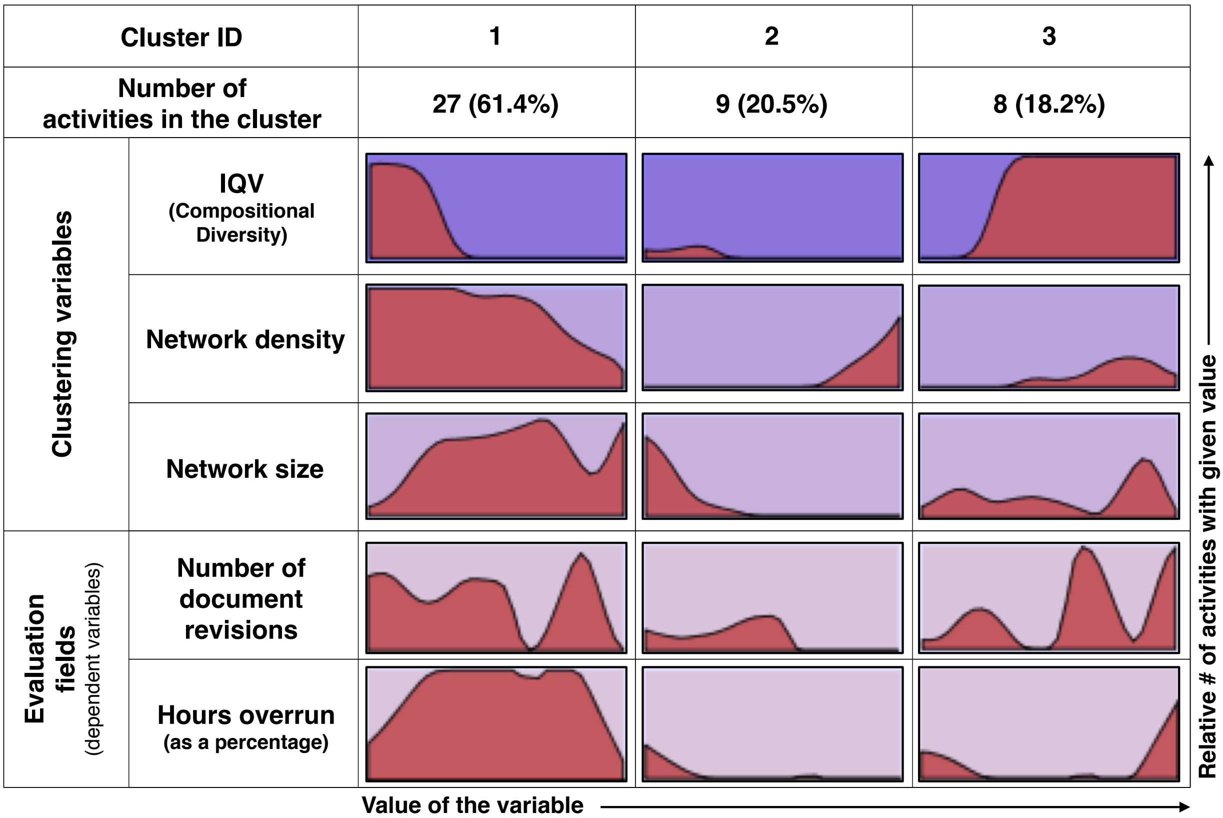


Figure 5-3: Graphical characterisation for the three identified clusters. Size, density and diversity of activities' organisation networks were utilised as clustering variables. For each cluster the relative distributions of clustering variables are shown as individual plots.

b) Examining performance differences across clusters through the one-way ANOVA test

To examine for significant performance differences between the clusters, I used a one-way ANOVA test. The test results showed no statistically significant difference in the average document revisions or the percentage of hours overrun across clusters. However, a statistically significant difference (p=0.05) was found between the standardised nominal amount of hours overrun and the clusters. ***This finding confirms that in this case study, the network characteristics of the clusters were associated with a statistically significant difference in the nominal amount of hours overrun***.

5.1.2 Discussion of case study results

The network structure and composition of 44 design activities were quantified through the application of the framework. The analysis of the relationship between the activities' characteristics and performance revealed three statistically significant clusters with distinctive combinations of size (numbers of people), density (connectivity between people), and IQV (compositional diversity). The characteristics of each cluster are as follows:

- Cluster 1: Most of the activities in the process (61.4%) tended to have low compositional diversity and low to medium density. Although performance metrics were average for this cluster, the relatively high number of people involved (average of 9 per activity) and low density (average of 59.21%) could pose higher coordination costs (Becker, 1992) and consequently, a decrease in information exchange.
- Cluster 2: The combination of activities with low heterogeneity among participants (average IQV of only 0.01), high connectivity (average density of 96.67%), and small size (average of only 3.33 participants) was associated with a consistently low average number of document reviews and the lowest percentage of hours overrun. In fact, activities in this cluster on average used 22% fewer hours than planned. Despite this good performance, previous literature indicates that low size, high density, and low heterogeneity may bring a higher risk of groupthink (Janis, 1982), leading to process isolation and communication problems between interdependent activities. In addition, the small organisational size of these activities leaves them more exposed to losing key knowledge if certain members are no longer involved.
- Cluster 3: This cluster was the most compositionally diverse group of activities by a significant amount (an average of 0.28 compared with the second highest of 0.03). At the same time, this cluster had the highest average number of document reviews (0.93) and the highest percentage of hours overrun (86%). This finding could indicate that activity performance suffered from communication challenges between activity members with different organisational affiliations, ultimately hindering the efficiency and effectiveness of the activity internal process.

Although we cannot directly identify causation through this analysis, the results of this case study indicated that activities with a low departmental diversity, high network density, and a small network size outperform other configurations. Why might this be the case? Designing a biomass power plant requires a combination of variant and adaptive designs, and strict regulations and strong technical interdependencies often hinder original design. As a result, we may speculate that a small, homogeneous group of well-connected individuals is more efficient in variant and adaptive design projects because deep technical expertise, coordination, and minimal communication barriers are required. This said, if causality exists, it also could be pointing in the opposite direction. For example ,more technically complex activities, inherently more likely to have performance problems, might be organised to include larger, more diverse groups of people. In any case, with this evidence the company can now run additional analyses to devise more definitive causal explanations, to take steps to leverage their best practices, and to intervene activities experiencing performance problems.

5.2 Networked interfaces

The analysis of the architecture of each process interfaces allowed company personnel to visualise how information passed between interdependent activities and to test how the characteristics of the organisation networks implementing each interface might affect the interface performance. If an architecture-performance relationship existed, such evidence could help the company identify best practices and explore in additional detail the root cause of any identified performance problems.

5.2.1 Application

At this level, the analysis was performed on interfaces between activity groups, each of which is associated with an individual subsystem. Unlike section 5.1, the analysis here was not performed between individual activities (see the process breakdown structure illustrated in 4.3.1 for additional details). The reason for this was the availability of sufficient information for both independent and dependent variables. For example, performance measures regarding interface problems were available only for interfaces between activity groups, not between individual activities.

Interface problems

Issues identified during the interviews as 'interface problems' were structured through the acquired process architecture and associated with specific process interfaces (see figure 5-2). Because only interface problems associated with the engineering design process were elicited, each interface problem between physical components could be traced to design process issues between activities belonging to different subsystems. Recurrent interface problems were related to one or more of the following aspects:

- **Required interfaces between components of different subsystems were not aligned or fully compatible due to technical specification issues.**

  For example, in the interface 'Air and flue gas → Steel-related activities', a problem was detected on the specifications of the steel supporting the air and flue gas subsystem.

- **Spatial clashes existed between parts or components belonging to different subsystems under development.**

  For example, in the interface problem 'Boiler and equipment design → Pressure parts design', spatial clashes were identified between grill tubes, pipes, and boiler equipment.

- **Information regarding technical specifications or procurement requirements, which should have been transferred between specific design activities, was missing.**

  For example, in the interface problem 'Pressure parts design → Procurement', the purchase order for a required a part was late, affecting the process schedule.

- **Other general misunderstandings or coordination issues** were identified between specific design activities that hindered the perceived performance of one activity in the interface.

Figure 5-2 shows all 79 process interfaces considered in this case, as well as the distribution of interfaces with or without problems (**Problem**: 15 – **No Problem, ✔**: 64). The process architecture was built as a binary-process DSM utilising the convention of inputs in columns (Eppinger and Browning, 2012, p. 5) and allowing interfaces and interface problems to exist in one or both directions; that is, the matrix represents a directed graph.

	Air and Flue Gas	Boiler and Equipment Design	Combustion System	COMOS Data	Electrical Control and Instrument.	External Piping	Load Plan and Layout	Overall Proj. Manag.	PFD + P&ID	Pressure Parts Design	Procurement	Design of steel structures
Air and Flue Gas				✔	✔		✔	✔	✔		✔	Problem
Boiler and Equipment Design	✔		✔	✔	✔	✔		✔		Problem		
Combustion System				✔	✔		Problem	✔			✔	
COMOS Data	✔		✔		✔	✔		✔	✔	✔		
Electrical Control and Instrumentation	✔		✔	✔		✔		✔	✔	Problem	✔	
External Piping				✔	✔		✔	✔	✔	Problem	Problem	Problem
Load Plan and Layout								✔				✔
Overall Project Management	✔	✔	✔	✔	✔	✔	✔		✔	Problem	Problem	✔
PFD + P&ID	✔	✔	✔	✔	✔	✔		✔		✔		
Pressure Parts Design				✔	Problem		✔	Problem	✔		Problem	
Procurement	✔		✔		✔	Problem		Problem		Problem		✔
Design of steel structures							✔	✔			✔	

Figure 5-2: Process design structure matrix: interfaces and interface problems. The matrix shows all 79 process interfaces identified using the convention of inputs in columns (for example, 'steel-related activities' require information from 'air and flue gas). Problem interfaces are labelled; those with no problem have a green checkmark.

Characterisation of Process Interfaces

Table 5-2 and figure 5-3 display descriptive statistics for the variables of size, density, and IQV, calculated for the 79 process interfaces. These descriptive statistics allow the variables to be treated as continuous, approximating a normal distribution, which is an important requirement for both the two-step clustering analysis and the one-way ANOVA test.

Table 5-2: Descriptive statistics for the network characteristics of the 79 process interfaces examined

	Minimum value	Maximum value	Mean	Std. Deviation
Size	15	37	25.53	5.523
Density (0 to 1)	.570	.805	.670	.549
IQV (0 to 1)	.461	.896	.698	.097

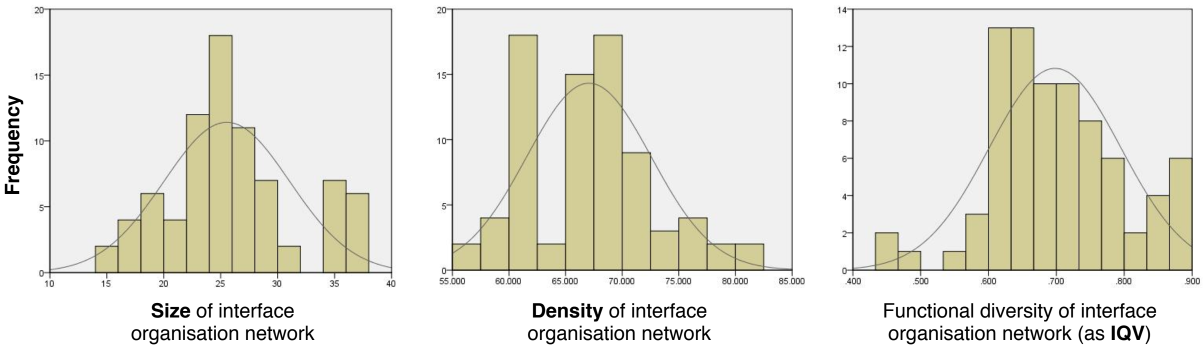
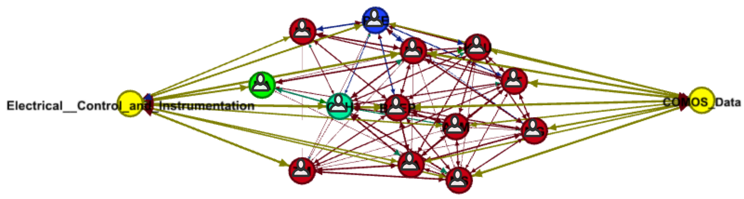
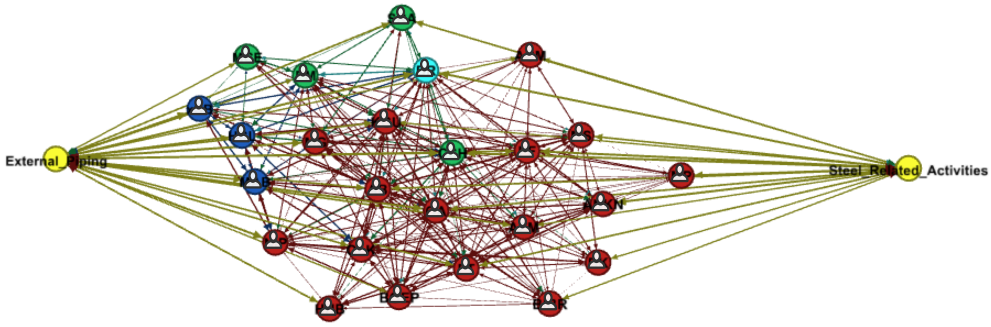


Figure 5-3: Histograms and approximate distribution curves for the network characteristics of size, density, and IQV of the 79 process interfaces examined. Size is expressed as an absolute value of the number of participants in the process interface, density as a percentage, and IQV on its 0-to-1 scale.

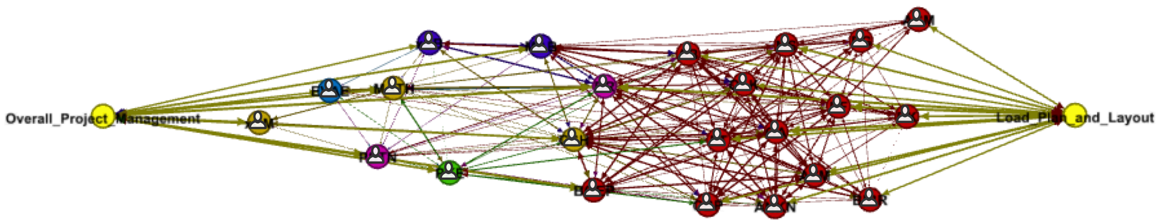
Figure 5-3 provides a graph-based characterisation for three of the 79 process interfaces, which were selected to illustrate various combinations of size, density, and IQV. Following the developed model, interfaces are represented as organisation networks between two activities and coloured to indicate their departmental affiliations.



A) Electrical Control and Instrumentation ⇔ Comos Data
Size: 15 – Density: 76% - IQV: 0.583



B) External Piping ⇒ Steel Related Activities
Size: 27 – Density: 68% - IQV: 0.625



C) Overall Project Management ⇔ Load Plan and Layout

Size: 25 – Density: 62% - IQV: 0.722

Figure 5-4: Three graphical examples of actual process interface characterisations. The graph layout is weighted and force-directed to represent different intensities of information exchanges. Edges map people to activities and people-people interactions. Project members are coloured according to their functional affiliation to a department. The far right and far left nodes represent the activities.

- Process interface A): Electrical Control and Instrumentation ⇔ Comos Data showed a relatively small, dense network with low diversity. Only one cohesive group was distinguishable.
- Process interface B): External Piping ⇒ Steel Related Activities was a larger, slightly sparser network. One cohesive group was still distinguishable; however, members from the same departments tend to group together.

- In contrast to the other two examples, process interface C): Overall Project Management ⇔ Load Plan and Layout was noticeable fragmented, with one cohesive and relatively homogeneous group (to the right of the graph) and a second sparse, cross-functional group to the left.

Employing the proposed method, I next applied (a) a two-step clustering analysis in which similar interfaces were grouped, and (b) a one-way ANOVA test to identify significance of the amount of interface problems across the identified.

Process Interface Analysis

a) Two-step clustering analysis

For the cluster analysis, the ‘distance measure’ used was log-likelihood, and the number of clusters was set to be determined automatically based on the Schwarz’s Bayesian information criterion (BIC). Three distinct groups with a ‘silhouette measure of cohesion and separation’ of 0.6 were obtained, which indicated good cluster quality (Tan, 2006).

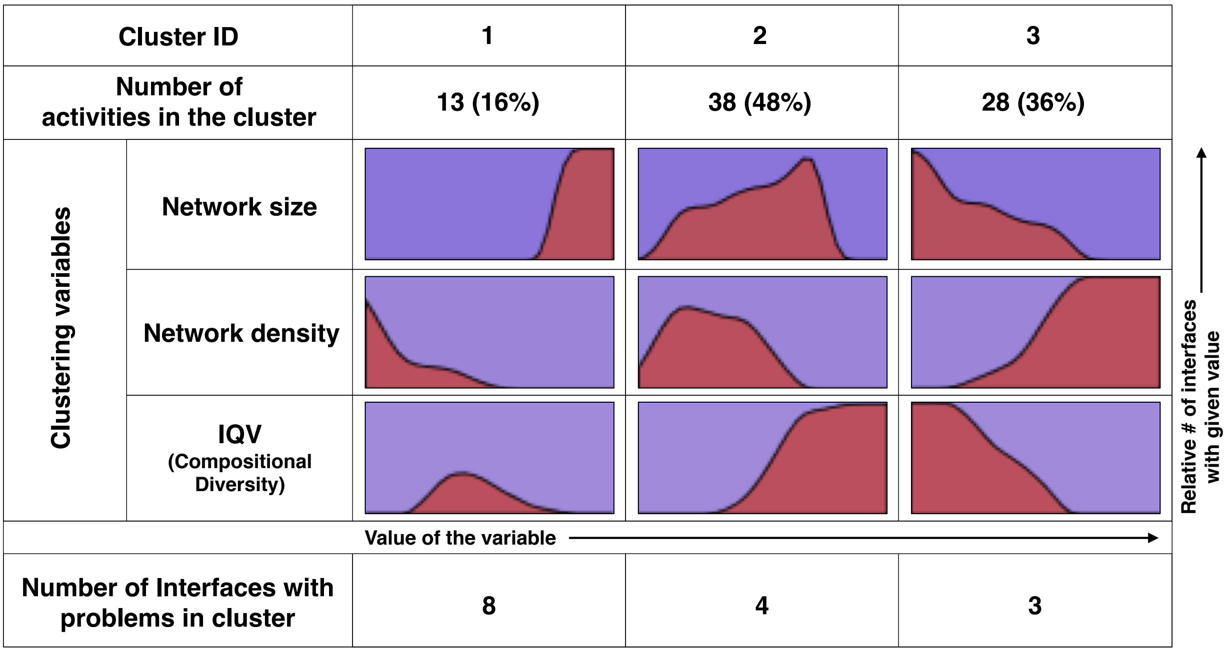


Figure 5-5: Graphical characterisation for the three identified clusters. The size, density, and diversity of the interfaces’ organisation networks were utilised as clustering variables. For each cluster, the relative distributions of clustering variables are shown as plots.

As shown in figure 5-5, the analysis revealed these cluster characteristics:

- Cluster 1 contained 16% of all interfaces, and the interfaces had a large number of people, low density, and medium heterogeneity. This cluster was composed mainly of interfaces between the design activity of 'pressure parts design' and other modular, integrative subsystem design activities. Despite having fewer interfaces than the others (16%), cluster 1 had the majority of the interface problems.

- Cluster 2 contained 48% of all interfaces, and the interfaces had a medium number of people, medium to low density, and high heterogeneity. This cluster was composed mainly of interfaces between integrative work (project management and procurement) and modular subsystem design activities.
- Cluster 3 contained 35% of all interfaces, and included those with a low number of people, high density, and low heterogeneity. This cluster was composed mainly of interfaces between modular and subsystem design activities. Clusters 2 and 3 had proportionally fewer interface problems compared with cluster 1.

b) Examining interface problem differences across clusters through the one-way ANOVA test

I used a one-way ANOVA test to understand if there were statistically significant differences in the proportion of interface problems between the clusters. The results indicated a highly significant ($p<0.01$) difference in the proportion of problems between the clusters, which confirmed the finding that the network characteristics of cluster 1 were associated with a greater likelihood of interface problems. Therefore, for this case study, ***interface problems were more likely to arise among interfaces with a larger number of participants and whose interactions had a relatively low density.***

In addition to the clustering and one-way ANOVA analysis, a logistic regression was performed to estimate the effects of network size, density, and IQV on the likelihood of an interface experiencing problems. The logistic regression model was statistically significant only for size, with $p < .005$ (0.22 coefficient), and explained 28.0% (Nagelkerke R^2) of the variance in interface problems. This revealed that network size was associated with an increased likelihood of exhibiting interface problems. However, unlike two-step clustering, logistic regression was unable to account for the effects of density and IQV.

5.2.2 Discussion of case study results

Interpretation of the case study results and practical implications for the company

Based on the cluster analysis and the potential effects of the characteristics of organisation network on performance reported in the literature, I propose a set of differentiated strategies to support project managers in each cluster:

- Cluster 1: The analysis indicated that interfaces found in this cluster, with interface networks higher in size and lower in density, were significantly more likely to have problems. From previous research (Burt, 1992; Chen and Gable, 2013; Tsai, 2001), we can infer that interfaces in this cluster were exposed to higher coordination costs, which could constrain information flows between activities. Strategies to mitigate these problems include: 1) increasing the organisational connectivity by incentivising more direct contact between the members of this interface, 2) allocating more resources to people who mediate interactions, because brokers (Gould and Fernandez, 1989) can increase efficiency of information exchanges, and 3) tearing down one or two activities at an interface to create two or more smaller interfaces with fewer people each, which has been found to be an effective way to manage complexity and improve modularity (Eppinger and Browning, 2012, p. 146; Steward, 1981).
- Cluster 2: Although reported interface problems were not high in this cluster, the main challenge was how to handle the interfaces' relatively high heterogeneity. Previous studies have shown that high functional diversity increases the likelihood of miscommunication and misalignment of objectives (Kleinsmann et al., 2007; Tushman et al., 1980). To mitigate these potential problems and benefit from the knowledge diversity inherent in heterogeneity, efforts should be made to ensure there are enough well connected individuals at the centre of the interface. These individuals should be able to bridge and translate different knowledge bases and align objectives, building capabilities to work across boundaries. In addition, as Maier, Kreimeyer, et al. (2009) suggested, a more reflective communication and overview that explicates each party's informational needs could be particularly helpful when dealing with cross-disciplinary interfaces.
- Cluster 3: Based on the relatively low number of reported problems, small size, high density, and low heterogeneity of the interfaces in this cluster, interface management here should be comparatively simpler. Nevertheless, these characteristics also may lead to groupthink and a lack of systemic perspective because of the narrower knowledge pool

available in this cluster (Burt, 1992; Easley and Kleinberg, 2010; Janis, 1982). As a result, additional, more diverse resources to increase heterogeneity could be beneficial, especially for interfaces dealing with central activities in the process that require a systemic perspective, or whose objectives are to produce innovative results.

5.3 Networked process

Following analyses of the design process at the design activity and process interface levels, we can turn to an analysis of the whole design process architecture of the case study. At the whole process level, the framework examines the comparison between planned and actual process architecture (section 5.3.1 and 5.3.2) and the relationship between the dynamic architecture of the whole design process and its planned design stages (sections 5.3.3 and 5.3.4).

5.3.1 Comparing the actual and the planned design process: application

Following the developed model and methods to compare the planned and actual process architectures (section 4.4), this section presents the results as applied to the main case study.

The framework was applied to the same 12 activity groups used to analyse interfaces (excluding on-site coordination). The data sources were:

For the planned design process architecture:

- List of tasks/activities acquired from workflow diagrams and activity record logs
- Groups of tasks/activities defined and refined using direct semi-structured interviews with the vice president of operations and the vice president of engineering
- Information dependencies between the 12 groups gathered through a structured interview using a valued process DSM matrix as a guide

For the actual design process architecture:

- The list of activities and activity groups gathered for the planned design process architecture (one-to-one mapping between activities and tasks)
- The process-organisation architecture gathered through an electronic questionnaire in which the 49 core team members registered their participation in activities, with participation weighted based on the reported degree of responsibility of each person
- The organisation architecture gathered through the same electronic questionnaire in which the 49 core team members registered their information exchanges with other project members, with interactions weighted based on reported frequency and impact.

The planned process architecture was directly derived from the valued process DSM matrix, normalised to a scale of 0 to 1 to be comparable. Figure 5-6 shows the resulting process architecture in matrix and graphic form.

Planned Process Architecture DSM

	Air and Flue Gas	Boiler and Equipment Design	COMOS Data	Combustion System	Electrical Control and Instrumentati	External Piping	Load Plan and Layout	Overall Project Management	PFD and P&ID	Pressure Parts Design	Procurement	Design of steel structures
Air and Flue Gas		0.0	0.7	0.0	0.3	0.0	1.0	0.3	0.7	0.0	0.7	0.7
Boiler and Equipment Design	1.0		0.7	1.0	1.0	1.0	0.0	0.3	0.0	1.0	0.0	0.0
COMOS Data	0.7	0.0		0.7	1.0	0.7	0.0	0.3	0.7	0.7	0.0	0.0
Combustion System	0.0	0.0	0.7		0.3	0.0	1.0	0.3	0.0	0.0	0.7	0.0
Electrical Control and Instrumentation	0.3	0.0	0.7	0.3		0.3	0.0	0.3	0.7	0.3	0.7	0.0
External Piping	0.0	0.0	0.3	0.0	0.3		1.0	0.3	0.3	0.3	0.7	0.7
Load Plan and Layout	0.0	0.0	0.0	0.0	0.0	0.0		0.7	0.0	0.0	0.0	1.0
Overall Project Management	0.3	0.3	0.3	0.3	0.3	0.3	0.3		0.3	0.3	1.0	0.3
PFD and P&ID	0.7	0.3	1.0	0.7	1.0	0.7	0.0	0.3		0.7	0.0	0.0
Pressure Parts Design	0.0	0.0	0.7	0.0	0.3	0.0	1.0	0.3	0.7		0.7	0.0
Procurement	0.3	0.0	0.0	0.3	0.3	0.3	0.0	0.3	0.0	0.3		0.3
Design of steel structures	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.3	0.0	0.0	0.7	

Planned Process Architecture Graph

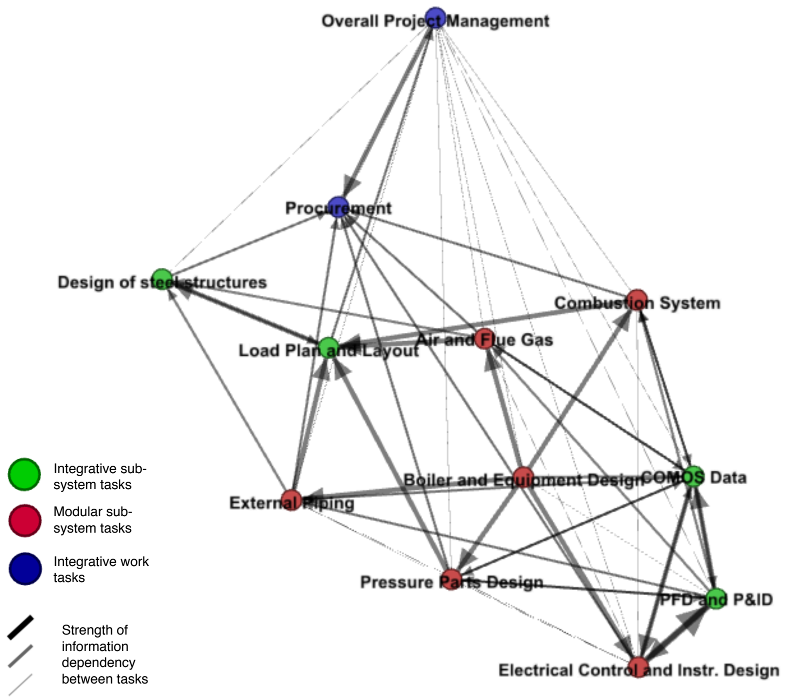


Figure 5-6: Planned process architecture in matrix and graph form. Matrix colours indicate relative strength of the information dependency between tasks, with red the lowest, green the highest.

The actual process architecture was derived from the intersection of process and organisation architectures, following the steps prescribed in section 4.4. For simplicity, figure

5-7 shows only the normalised results corresponding to the portion of the information centrality nearness matrix that shows the final output in terms of activity-activity information flows.

Actual Process Architecture
(DSM derived from information centrality nearness matrix)

	Air and Flue Gas	Boiler and Equipment Design	COMOS Data	Combustion System	Electrical Control and Instrumentati	External Piping	Load Plan and Layout	Overall Project Management	PFD and P&ID	Pressure Parts Design	Procurement	Design of steel structures
Air and Flue Gas		0.78561	0.697176	0.556801	0.594094	0.739947	0.778301	0.351718	0.667871	0.62801	0.607415	0.753809
Boiler and Equipment Design	0.78561		0.469144	0.562007	0.398506	0.78988	0.826056	0.324452	0.652475	0.757179	0.521436	0.792054
COMOS Data	0.697176	0.469144		0.680703	0.80893	0.399806	0.518993	0.213024	0.689574	0.379395	0.533762	0.41023
Combustion System	0.556801	0.562007	0.680703		0.826832	0.487318	0.469729	0.144861	0.762966	0.359029	0.501339	0.462963
Electrical Control and Instrumentation	0.594094	0.398506	0.80893	0.826832		0.325782	0.305268	0.245593	0.652008	0.260198	0.473528	0.320271
External Piping	0.739947	0.78988	0.399806	0.487318	0.325782		0.666873	0.370546	0.619292	0.801822	0.628187	0.714968
Load Plan and Layout	0.778301	0.826056	0.518993	0.469729	0.305268	0.666873		1E-07	0.499067	0.721749	0.545525	1
Overall Project Management	0.351718	0.324452	0.213024	0.144861	0.245593	0.370546	1E-07		0.37463	0.265308	0.225484	0.103766
PFD and P&ID	0.667871	0.652475	0.689574	0.762966	0.652008	0.619292	0.499067	0.37463		0.540695	0.603507	0.56236
Pressure Parts Design	0.62801	0.757179	0.379395	0.359029	0.260198	0.801822	0.721749	0.265308	0.540695		0.640737	0.750774
Procurement	0.607415	0.521436	0.533762	0.501339	0.473528	0.628187	0.545525	0.225484	0.603507	0.640737		0.52456
Design of steel structures	0.753809	0.792054	0.41023	0.462963	0.320271	0.714968	1	0.103766	0.56236	0.750774	0.52456	

Actual Process Architecture Graph

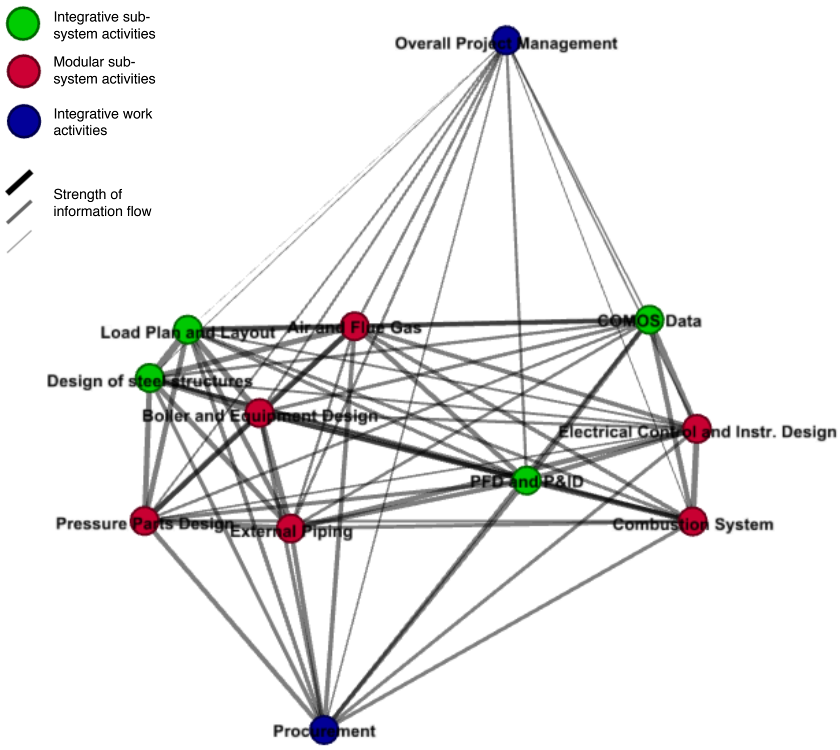


Figure 5-7: Actual process architecture in matrix (normalised values) and graph form. Matrix colours indicate relative strength of the information flow between activities, with red the lowest, green the highest.

These two results were informative in their own right, providing aggregated pictures of all planned information dependencies and actual information flows that can be used to increase process overview. However, the most useful value emerged when the planned and actual architectures were directly compared, which was possible because these architectures were expressed in compatible forms. Figure 5-8 shows the results of subtracting the planned process DSM from the actual process architecture, to generate a Δ design process matrix in which negative differences indicate lower than expected information flows and positive values indicate an information flow surplus, in relative terms.

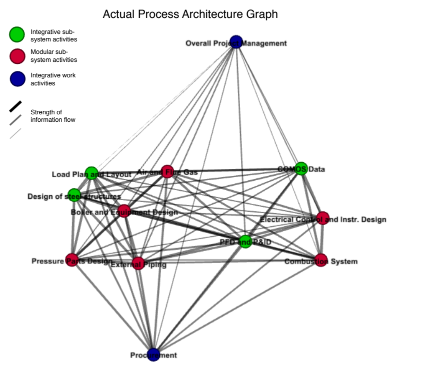
Actual Process Architecture

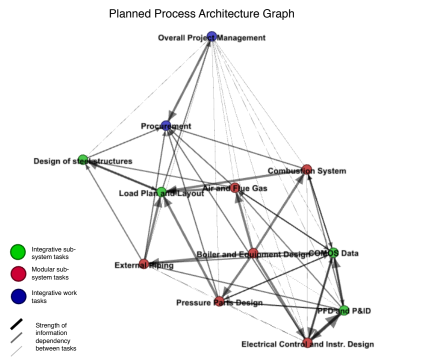
	Air and Flue Gas	Boiler and Equipment Design	COMOS Data	Combustion System	Electrical Control and Instrumentati	External Piping	Load Plan and Layout	Overall Project Management	PFD and P&ID	Pressure Parts Design	Procurement	Design of steel structures
Air and Flue Gas		0.78561	0.697176	0.556801	0.594094	0.739947	0.778301	0.351718	0.667871	0.62801	0.607415	0.753809
Boiler and Equipment Design	0.78561		0.469144	0.562007	0.398506	0.78988	0.826056	0.324452	0.652475	0.757179	0.521436	0.792054
COMOS Data	0.697176	0.469144		0.680703	0.80893	0.399806	0.518993	0.213024	0.689574	0.379395	0.533762	0.41023
Combustion System	0.556801	0.562007	0.680703		0.826832	0.487318	0.469729	0.144861	0.762966	0.359029	0.501339	0.462963
Electrical Control and Instrumentation	0.594094	0.398506	0.80893	0.826832		0.325782	0.305268	0.245593	0.652008	0.260198	0.473528	0.320271
External Piping	0.739947	0.78988	0.399806	0.487318	0.325782		0.666873	0.370546	0.619292	0.801822	0.628187	0.714968
Load Plan and Layout	0.778301	0.826056	0.518993	0.469729	0.305268	0.666873		1E-07	0.499067	0.721749	0.545525	1
Overall Project Management	0.351718	0.324452	0.213024	0.144861	0.245593	0.370546	1E-07		0.37463	0.265308	0.225484	0.103766
PFD and P&ID	0.667871	0.652475	0.689574	0.762966	0.652008	0.619292	0.499067	0.37463		0.540695	0.603507	0.56236
Pressure Parts Design	0.62801	0.757179	0.379395	0.359029	0.260198	0.801822	0.721749	0.265308	0.540695		0.640737	0.750774
Procurement	0.607415	0.521436	0.533762	0.501339	0.473528	0.628187	0.545525	0.225484	0.603507	0.640737		0.52456
Design of steel structures	0.753809	0.792054	0.41023	0.462963	0.320271	0.714968	1	0.103766	0.56236	0.750774	0.52456	

−

Planned Process Architecture DSM

	Air and Flue Gas	Boiler and Equipment Design	COMOS Data	Combustion System	Electrical Control and Instrumentati	External Piping	Load Plan and Layout	Overall Project Management	PFD and P&ID	Pressure Parts Design	Procurement	Design of steel structures
Air and Flue Gas		0.0	0.7	0.0	0.3	0.0	1.0	0.3	0.7	0.0	0.7	0.7
Boiler and Equipment Design	1.0		0.7	1.0	1.0	1.0	0.0	0.3	0.0	1.0	0.0	0.0
COMOS Data	0.7	0.0		0.7	1.0	0.7	0.0	0.3	0.7	0.7	0.0	0.0
Combustion System	0.0	0.0	0.7		0.3	0.0	1.0	0.3	0.0	0.0	0.7	0.0
Electrical Control and Instrumentation	0.3	0.0	0.7	0.3		0.3	0.0	0.3	0.7	0.3	0.7	0.0
External Piping	0.0	0.0	0.3	0.0	0.3		1.0	0.3	0.3	0.3	0.7	0.7
Load Plan and Layout	0.0	0.0	0.0	0.0	0.0	0.0		0.7	0.0	0.0	0.0	1.0
Overall Project Management	0.3	0.3	0.3	0.3	0.3	0.3	0.3		0.3	0.3	1.0	0.3
PFD and P&ID	0.7	0.3	1.0	0.7	1.0	0.7	0.0	0.3		0.7	0.0	0.0
Pressure Parts Design	0.0	0.0	0.7	0.0	0.3	0.0	1.0	0.3	0.7		0.7	0.0
Procurement	0.3	0.0	0.0	0.3	0.3	0.3	0.0	0.3	0.0	0.3		0.3
Design of steel structures	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.3	0.0	0.0	0.7	





Δ Design Process Matrix Actual - Planned

=

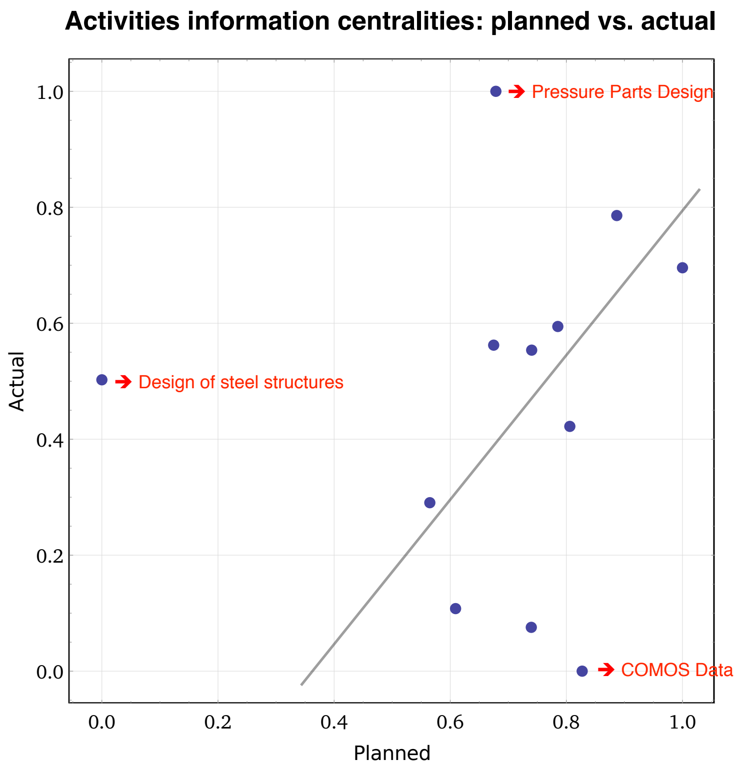
	Air and Flue Gas	Boiler and Equipment Design	COMOS Data	Combustion System	Electrical Control and Instrumentati	External Piping	Load Plan and Layout	Overall Project Management	PFD and P&ID	Pressure Parts Design	Procurement	Design of steel structures
Air and Flue Gas		0.786	0.031	0.557	0.261	0.740	-0.222	0.018	0.001	0.628	-0.059	0.087
Boiler and Equipment Design	-0.214		-0.198	-0.438	-0.601	-0.210	0.826	-0.009	0.652	-0.243	0.521	0.792
COMOS Data	0.031	0.469		0.014	-0.191	-0.267	0.519	-0.120	0.023	-0.287	0.534	0.410
Combustion System	0.557	0.562	0.014		0.493	0.487	-0.530	-0.188	0.763	0.359	-0.165	0.463
Electrical Control and Instrumentation	0.261	0.399	0.142	0.493		-0.008	0.305	-0.088	-0.015	-0.073	-0.193	0.320
External Piping	0.740	0.790	0.066	0.487	-0.008		-0.333	0.037	0.286	0.468	-0.038	0.048
Load Plan and Layout	0.778	0.826	0.519	0.470	0.305	0.667		-0.667	0.499	0.722	0.546	0.000
Overall Project Management	0.018	-0.009	-0.120	-0.188	-0.088	0.037	-0.333		0.041	-0.068	-0.775	-0.230
PFD and P&ID	0.001	0.319	-0.310	0.096	-0.348	-0.047	0.499	0.041		-0.126	0.604	0.562
Pressure Parts Design	0.628	0.757	-0.287	0.359	-0.073	0.802	-0.278	-0.068	-0.126		-0.026	0.751
Procurement	0.274	0.521	0.534	0.168	0.140	0.295	0.546	-0.108	0.604	0.307		0.191
Design of steel structures	0.754	0.792	0.410	0.463	0.320	0.715	0.333	-0.230	0.562	0.751	-0.142	

Figure 5-8: The actual minus the planned process architecture to obtain a Δ design process matrix. Actual and planned matrices are normalised (scale zero to one) and all cells coloured to reflect the relative strength of the information dependency or flows: red lowest, green highest.

One strategy to quantify the extent of alignment between the planned and actual architectures is to run a linear regression each having the values of one, and using the output produced by the information centrality nearness matrix (this requires calculating the information centrality nearness matrix for the planned process architecture). Such an analysis can be applied at the activity or activity group levels using Stephenson and Zelen's (1989) information centrality algorithm, or at the process interface level using Stephenson and Zelen's (1989) information centrality nearness matrix. If the computed actual process architecture has a meaningful relationship to the planned process architecture (and assuming the design process examined does not have abnormal behaviour), the regression should show a positive correlation between the actual and the planned process architectures. In other words, we can reasonable expect that on average the information exchanges between individuals will correspond to the information dependencies between the activities in which they participate. The highest deviations from the regression line can be taken as outliers for further examination. Any outlier above the regression line (assuming 'actual' is plotted on the vertical axis) represents higher information flow than expected. In turn, any outlier below the regression line represents lower information flow than expected and might be a sign of insufficient informational connectedness.

At the level of activity groups, the results shown in figure 5-9 indicate the degree of alignment of each activity group to its expected information centrality. Above the regression line, the most important outliers were the activity groups, 'Design of steel structures' and 'Pressure parts design'. The activity group, 'COMOS data', was below the regression line. Discarding these three outliers, the information centralities of the actual and the planned process exhibit a positive correlation and coincide with the assumption about overall alignment between information dependencies and information flows, although the sample size was very small.

The results at the level of interfaces between activity groups, shown in figure 5-10, indicated the degree of alignment of each interface between activity groups to their expected information centrality. Below the regression line, the interfaces with the highest misalignment were 'Load Plan and Layout – Overall Project Management', 'Overall Project Management – Design of Steel Structures' and 'Load Plan and Layout – Overall Project Management'. Now with a bigger sample size and without discarding outliers, we can confirm with 99% confidence that a positive correlation between planned and actual architecture exists.



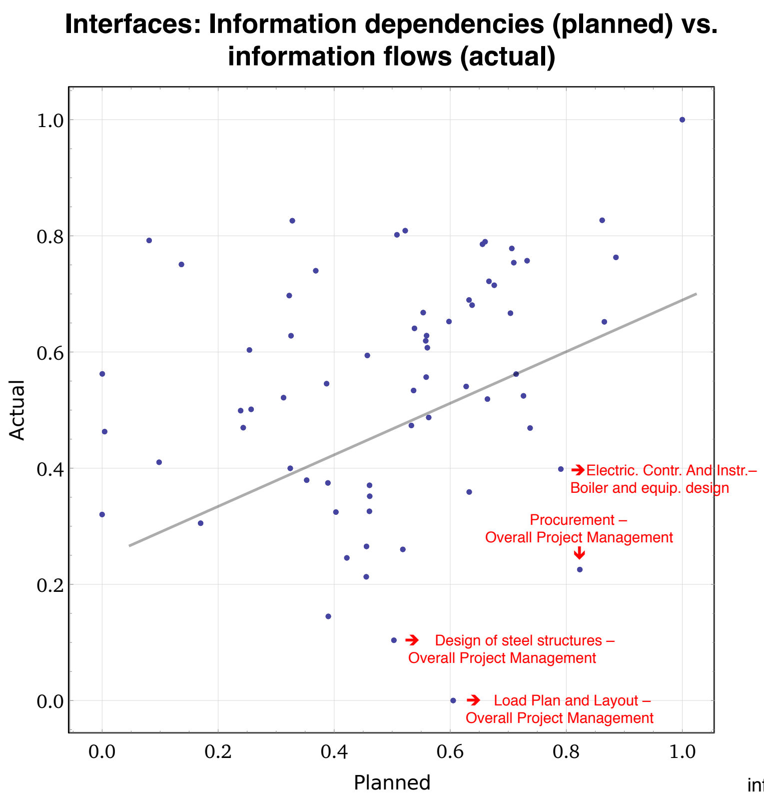
Activities sorted by information centrality delta	Delta actual-planned
COMOS Data	-0.8272394
Electrical Control and Instr	-0.664038
Overall Project Management	-0.501313
Load Plan and Layout	-0.383844
Boiler and Equipment Design	-0.3042967
Combustion System	-0.274327
Air and Flue Gas	-0.1908747
PFD and P&ID	-0.1866497
Procurement	-0.112573
External Piping	-0.1010766
Pressure Parts Design	0.32140364
Design of steel structures	0.50246001

Regression parameters after discarding three outliers:

	estimate	confidence
α	0.5904 ± 0.0766	$> 99\%$
β_1	0.3652 ± 0.1496	$> 95\%$

After discarding three outliers (design of steel structures, pressure parts design, and COMOS data) actual and planned information centralities are positively correlated with 95% of confidence

Figure 5-9: Regression between the values of actual and planned information centrality at the level of activity groups.



Interfaces with the largest misalignments between information dependencies and actual information flows	Delta actual-planned
Load Plan and Layout-Overall Project Management	-0.605082
Procurement-Overall Project Management	-0.597838
Design of steel structures-Overall Project Management	-0.399139
Electrical Control and Instr-Boiler and Equipment Design	-0.392116
Pressure Parts Design-COMOS Data	-0.273674
Boiler and Equipment Design-Combustion System	-0.268498
Electrical Control and Instr-Pressure Parts Design	-0.258105

Regression parameters

	estimate	confidence
α	0.3419 ± 0.0534	$> 99\%$
β_1	0.2895 ± 0.0926	$> 99\%$

Planned information dependencies and actual information flows are positively correlated with a 99% confidence (*t* test including all points)

Figure 5-10: Regression between the values of the actual and the planned information centrality nearness matrix at the interface level.

5.3.2 Comparing the actual and the planned design process: discussion

The previous high-level aggregated view of the actual and planned process architectures (by activity group) allowed a side-by-side comparison of what managers planned based on information dependencies between tasks against the actual information flows based on information exchanges between people performing activities. As a result, this case study contrasts the planned top-down view of the process against the bottom-up actual way in which the architecture was built through the participants' actions, something that previously was not possible.

Reviewing the Δ design process matrix, we can identify specific instances in which the framework detected information dependencies that were addressed with a volume of information relatively lower than expected given the strength of the information dependency. Some of the highest mismatches between planned information dependency and actual information flow were at the interface between 'Overall Project Management' and 'Procurement' (-0.75), 'Load Plan and Layout' and 'Overall Project Management' (-0.667), and 'Boiler and Equipment Design' and 'Electrical Control and Instrumentation' (-0.601). We also see this same result in figures 5-6 and 5-7 when comparing the expected versus the actual distances between the nodes representing activity groups; for example, the distance between 'Procurement' and 'Overall Project Management' is noticeable longer in the actual design process architecture.

Two relevant questions are: How do these results connect with actual problems the company faced? And, how can the company use this information to improve its design process? Before examining the actual results, an important aspect that should be taken into account is that misalignments are not intrinsically undesirable. For example, the planned process architecture could have underestimated or overestimated an information dependency, in which case it would not be ideal to match the dependency with an equal level of information flow. Therefore, misalignments at the level of each activity or interface cannot be immediately taken to indicate a problem. However, **on average at the level of the whole process, information dependencies should be addressed with a matching level of information flow.**

In this case, the overall planned architecture reflected well the actual architecture, with a few exceptions. One of the most noticeable exceptions was the interface between 'Overall Project Management' and 'Procurement'. Coincidentally, during the interface analysis, that particular interface was indeed reported to have inadequate information flow. Now, the framework has provided good data-driven arguments to believe that one reason for this problem lay in the insufficient information flow in this identified interface. We can use insights derived from the Δ design process matrix to find ways to adjust the actual information flows without negatively affecting the overall architecture or any particular activity. An opportunity for this adjustment was found in the information flow surplus between 'Overall Project Management' and 'PFD and P&ID'. The analysis did not identify a problem in this interface, which would indicate that efforts could be redirected from this interface to 'Procurement'. In practical terms, members of these three activity groups could be informed about the situation and asked to redirect efforts accordingly. Other alternatives may include locating people working at 'Overall Project Management' and 'Procurement' closer together, increasing the distance from those working on 'PFD and P&ID', or restructuring meetings to redirect and/or strengthen information interactions.

5.3.3 The dynamic architecture of the whole design process: application

This section presents the results of applying temporal network analysis to the information network in the descriptive case study. The data sources for the analysis are summarised and the overall information flow network topology is presented by stage. Next, the results of the evolving design process at the level of activity categories are discussed. Finally, centralisation patterns for the whole information network are calculated, which allows exploration of the evolution of the overall information centralisation.

Organisation domain data: Data with which to map the organisation domain included information exchanges between members of the engineering design project, spanning 15 departments. The exchanges between the participants were reported directly via an electronic questionnaire, individually answered by the 49 core project members. Selecting from among 77 current and former project members with engineering design responsibilities, respondents indicated if they had had any information exchange with any of the listed employees. They also quantified their information exchange interactions according to frequency, impact, context, and the originator of the interaction. The questionnaires yield a total of 756 information exchange pairs (dyads).

Because only 49 of the 77 employees with engineering design responsibilities were selected to answer the questionnaire, in some cases only one side of the dyad reported on the interaction. To ensure consistency, I symmetrised the information exchange matrix (using maximum value across each dyad), with the assumption that interactions may not have been reported because the employee did not recall them or because one of the participants was not selected as a questionnaire respondent. Cases in which neither party in a potential dyad took the survey were individually examined with project leaders to ensure important information exchanges were not missed.

Process domain data: Data about the process domain included a detailed list of project activities (used internally by the company for project management and reporting) and their information dependencies. After eliminating non-design activities, 148 activities were determined to be suitable for the dynamic network analysis. This final list was validated through the company's technical documentation, which included workflow diagrams and Gantt charts, and interviews with the vice president of operations, vice president of engineering, and the project manager.

With the help of company engineers, the activities were categorised into the 13 activity groups[5] listed in table 5-3. This first level of categorisation was based on the identification of cohesive work packages related to the subsystems under development or other common characteristics among the activities. To identify the planned relationships across the 13 activity groups, I created a DSM based on information dependencies revealed by the project managers and the existing workflow diagrams. This DSM analysis enabled comparison of information dependencies with actual information flows and placement of the activity groups into one of three activity categories: integrative work activities, integrative subsystem activities, or modular subsystem activities.

Cross-domain mapping data: Data for mapping the process and organisation domains was obtained through company records that indicated each time a project member performed one of the 148 activities. The person performing the activity placed this information directly into a database at least weekly, logging the date of the activity and hours invested. Project managers used these reports to track resources and update the project budget and schedule. With this dataset of 11,742 records, and the information about the organisation domain, I was able to identify the possible pathways of information flow over time. Although the project spanned more than two years, temporal data for all domains was aggregated on months to reduce the noise of daily fluctuations while retaining sufficient temporal detail.

[5] Due to the relevance in the process dynamics of the activity group's 'on-site coordination', this activity group was added to the list of 12 activity groups used in the previous analysis.

Other considerations: I applied three rules to embed temporal information into the organisation domain, which in this case was based on reported information exchanges. First, for an information exchange to exist between two parties, those parties should have previously logged at least one activity in the project. Second, the earliest possible date of the information exchange was the later date of each party's first logged activity. Third, once an exchange existed, further exchanges would occur any time both parties logged activity in the same period of analysis. Although these rules contained assumptions about the timing and context of information exchanges, they seemed to be a fair, albeit simplified, representation of the information exchange dynamics, based on the company's direct validation. Future studies could improved and complement this process with an analysis of information exchanges via email or other information systems that might already include a timestamp as part of the metadata.

Finally, to implement the dynamic network analysis, I computed the measures of betweenness centrality and group betweenness centralisation, utilising the Condor software package (Gloor, 2013). The computation followed the approach of the temporal communication flow structure proposed by Gloor and Zhao (2004), but adapted to the networked process framework.

Table 5-3: Table of all the activity groups considered for the dynamic analysis at the whole process level

Activity categories (A, B, C) and activity groups (A1–A3, B1–B4, and C1–C6)

A: Integrative work activities	B: Integrative subsystem activities	C: Modular subsystem activities
A1: Overall project management A2: Procurement A3: On-site coordination	B1: Design of steel structures B2: Load plan and layout B3: Process flow diagram (PFD) + piping and instrumentation diagram (P&ID) B4: COMOS (database-related work)	C1: Boiler and equipment design C2: External piping design C3: Pressure parts design C4: Air and flue gas design C5: Combustion system design C6: Electrical, control, and instr. design

Overall information network structure

The model and methods previously introduced to quantify dynamic information flows between activities were applied to the whole network. Figure 5-9 shows the results of the model in four graphs, one per design stage. Each node represents one of the 13 activity groups used to analyse networked interfaces, plus one additional group related to on-site coordination. The edges connecting each activity group show the weighted information flows between them. Because the model considers all possible information paths between each pair of activities, the graphs tend to be almost fully connected. As such, the structure depends mainly on the weights of the edges. To ease interpretation, the nodes and the edges have been sized based on their cumulative information centralisation and displayed in the graph using a weighted force-directed layout.

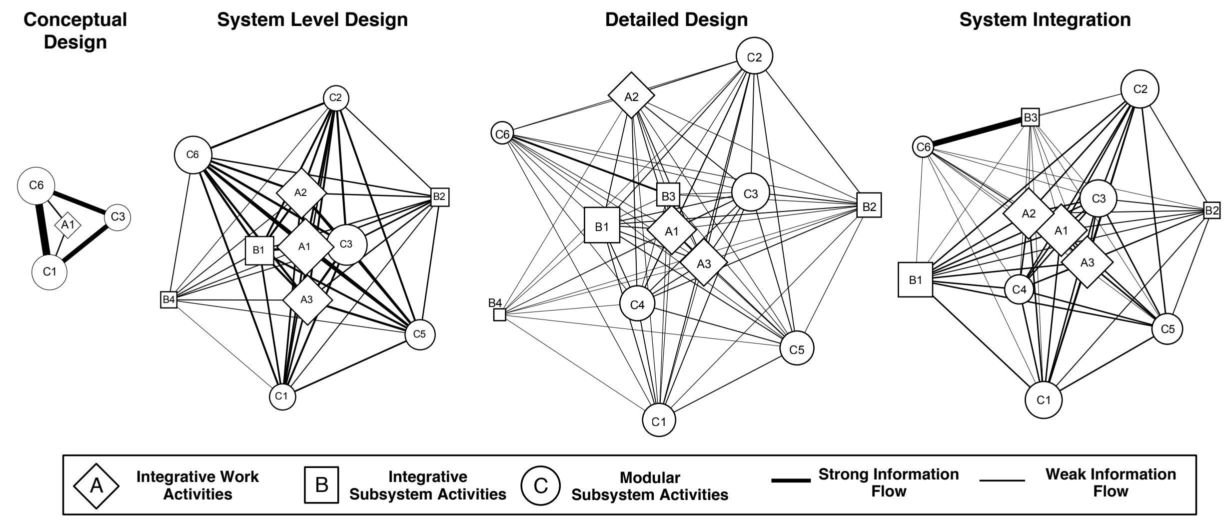


Figure 5-11: Information network for each of the analysed stages, showing activity groups connected by their weighted information flows. Node size reflects relative information centrality, and edge weight the relative amount of information flow.

From the graphs in Figure 5-11, we can observe that the distribution of information centrality tends to follow the expected structure of the network at each stage summarised in table 4-4.

- In the conceptual design stage, although overall project management (A1) did not centralise information, the relative composition and size of the information network followed the prediction.
- In the system level design stage the network grew, but maintained a relatively centralised structure with overall project management (A1) adopting a more dominant information centralisation profile. Despite representing only 3 of 11 activity groups, the combination of A1 with the two other integrative work activities, A2 and A3, accounted for a large part of the information centrality in this stage (37%).
- In the detailed design stage, the network grew once again, reaching its maximum number of activity groups (13). Also, the structure at this stage became more distributed, which is reflected in a lower proportion of information centrality held by integrative work activities (26%) and a corresponding increase in information centrality held by modular (52%) and integrative subsystem activities (20%).

- In the system integration stage integrative work activities regained centrality (31%). Also the coefficient of variation (mean/standard deviation) of the weighted graph density increased from 0.26 (detailed design) to 0.44, showing evidence of an overall increase in information centralisation.

Information centrality across activity categories

For each of the three activity categories, information centrality was calculated monthly (figure 5-12) in order to identify how closely information centrality matched the idealised value per stage. To obtain information centrality, I computed betweenness centrality for each category of activities. In the results, the expected patterns indeed emerged for the evolution of information centrality between the three activity categories, with the exception of conceptual design. System level design began with integrative work holding high information centrality, which declined over time as the modular subsystem activities entered during this stage. Detailed design was dominated by the development of the modular and integrative subsystems, with a sharp decrease in centrality by integrative work activities. In turn, during the system integration stage, centrality by activities related to integrative work increased, integrative subsystems remained at the levels of detailed design, and modular subsystems decreased their centrality over time.

In the conceptual design stage, information centrality alternated between integrative work and modular subsystem activities, whereas only the former was expected. As an explanation for this pattern, the company said it had extensive prior experience in these kinds of projects, which allowed key technical areas (a few modular subsystem activities) to lead during the conceptual stage of the project; thus, the deviation from the expected pattern was a natural consequence of that company's particular set-up.

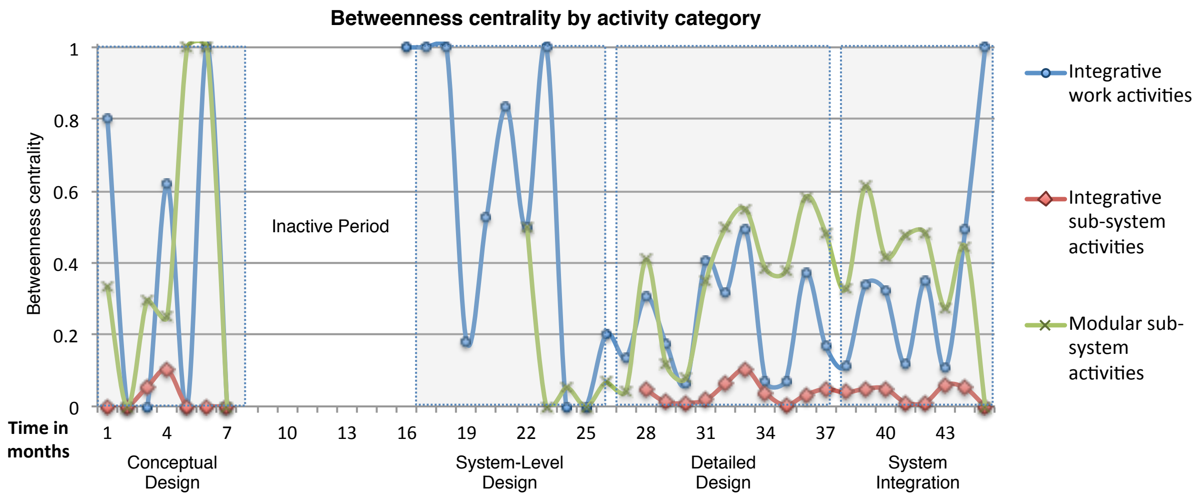


Figure 5-12: Evolution of information centrality across activity categories, measured using betweenness centrality (BC)

Evolution of information centralisation at the project level

To obtain the evolution of information centralisation at the project level, I calculated group betweenness centralisation (GBC) for each month of the project. Unlike betweenness centrality, which is a node-level measure, GBC describes the whole network centralisation and can be interpreted as a measure of the distribution of information centrality among different activities (and consequently, subsystems) in the whole project. A high GBC indicates that only a few activities hold most of the information centrality, and therefore, information flows tend to be more centralised. The lowest GBC (0) indicates that information centrality is evenly distributed and can be interpreted as a sign of high process modularity and relative autonomy between the subsystems under development. Despite oscillations in the measures (partially due to periods of inactivity), figure 5-13 shows evidence of patterns that matched the expected evolution of GBC at each stage of the project. Conceptual design was characterised by only a few activities holding most of the information centrality and coordinating inputs from multiple areas. System-level design exhibited a similar pattern, which decreased as the detailed design was about to start. The detailed design showed signs of increased process modularity because of the high technical specialisation and detailed work (reflected in its low GBC score). Finally, system integration showed a rising GBC score, a sign of the need for higher levels of coordination to complete integration of the different subsystems at the end of this stage.

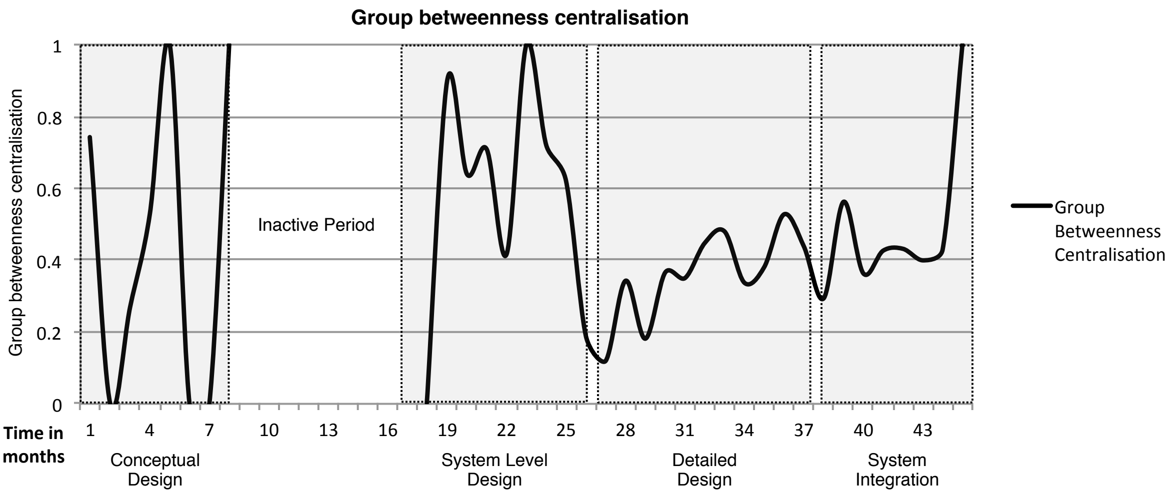


Figure 5-13: Evolution of information centralisation, measured using group betweenness centralisation (GBC)

5.3.4 The dynamic architecture of the whole design process: discussion

What are the implications of the changes in information centrality and centralisation for engineering design theory and practice?

The application of this part of the framework identified patterns in information centrality and centralisation associated with different stages of the design process. As described in the discussion of 'patterns instead of models' (Clarkson and Eckert, 2005), the means to identify such patterns of designing are crucial to understanding the actual process and to uncovering causal explanations. In contrast, models designed to provide abstract descriptions of generic design processes can be used as a basis of comparison and for interpreting patterns.

In the case study, the emergence of meaningful and interpretable patterns from the dynamic analysis of more than 40 periods and thousands of valued dyads served as positive proof-of-concept for the framework. Moreover, based on these empirical results, the claim can be made that the discovered information flow patterns were related to the project's progression, and consequently, can be compared with idealised models to identify and correct unexpected and potentially undesirable information flow patterns.

The observed information flow patterns also allowed a meaningful macro-level categorisation of activities into three classes, based on their distinctive information centrality patterns and evolution. We could distinguish among modular subsystem, integrative subsystems, and integrative work design activities based on not only company insights, observations, and static network models, but also their characteristic network dynamics. This ability to categorise activities can allow researchers to perform simplified analysis: Instead of following the dynamics of each activity or activity group, they need only study the patterns of three activity categories to visualise a meaningful distribution of the information centrality linked to SE-V model stages.

5.4 Chapter summary

This chapter provided a comprehensive proof-of-concept for the overall proposed framework, applying it to the three levels of analysis and comparing the planned process architecture against the actual process architecture. The key takeaways are:

- The framework developed in chapter 4 can be applied in practice at each level of analysis to produce the expected outputs.
- The required information sources are relatively easy to gather, and in many instances, are already fully or partially available inside a company as information produced during the design process.
- Although for the most part the three levels of analysis require the same data, the results of applying the framework at each level reveal distinct insights, confirming the need for a multilevel approach.
- The company reported that the results obtained from the analysis seemed to appropriately reflect the actual process architecture, which provides evidence that the results achieved face validity from the receiving company's perspective.
- The results were considered not only a good reflection of decision makers' knowledge, but also expanded and connected previously unavailable insights.

A summary to guide the application of the NPr Framework is available in appendix H in the form of a graphical workflow.

6 EVALUATION AND DISCUSSION

Nothing has such power to broaden the mind as the ability to investigate systematically and truly all that comes under thy observation in life

—Marcus Aurelius

In section 4.6, I described how the NPr Framework could indeed answer each of the 10 research sub-questions. Similarly, chapter 5 provided a complete proof-of-concept for the framework's application, discussing results obtained at each level of analysis. The case study indicated that not only could the NPr Framework be implemented in the design process of a real engineering system, but also the data-driven support the framework provides could produce new and valuable insights for the company. In this chapter I provide an overall evaluation and discussion of the framework as a whole: First, I elaborate on the key assumptions behind the framework (6.1), and then review the most important limitations (6.2). In section 6.3, I evaluate the framework's outputs for industry and research in relation to the previously defined success criteria. Finally, in section 6.5 I reflect on lessons learned during the course of this research project.

6.1 Assumptions

Characterising design as a socio-technical network

The conventional assumption is that the design process of engineering systems is a complex socio-technical system of information transformation (Chira, 2005; Culley, 2014; Hubka, 1996; Shears, 1971) that can be characterised as a network. This system includes in the product domain interconnected components that must be designed, in the organisation domain interconnected people participating in the engineering design, and in the process domain activities and tasks (Browning, 2002; Eppinger and Salminen, 2001; Maurer, 2007).

What is not a conventional assumption, at least from the point of view of research in process domain architecture, is that task networks and activity networks are structurally and compositionally different from each other. In this thesis, task networks in the design process are exclusively associated with technical information dependencies, while activity networks are exclusively associated with social information flows. Moreover, the proposed model treats each task and each activity as a network in its own right. Tasks exist at the intersection of the architectures of the process and the product domains. In turn, activities exist at the intersection of the architectures of the process and the organisation domains. For tasks, the model takes as a reference the Signposting Framework (Clarkson and Hamilton, 2000; Wynn et al., 2006) in which tasks are connected indirectly through parameters that can be assigned to specific components. For activities, the model extrapolates findings from previous research on collaboration and change propagation in design (e.g. Christian 1995; Durugbo et al. 2011; Pasqual & de Weck 2011), in which information exchanges between people are integrated with the process domain through their participation in activities. However, the activity models surveyed were insufficient to respond to all the research objectives, and therefore, I developed a new framework to characterise the actual architecture of the design process.

Complexity, networks and emergence in the design process of large projects

The proposed conceptual model relies on the idea that the design process, when modelled as a network, yields new and valuable information about the design process that can provide understanding and support (e.g. Kreimeyer & Lindemann 2011). More specifically, the model uses the premise that the non-linear properties and emergent behaviour of complex systems (Clarkson and Eckert, 2005, chap. 7 p.180; Holland, 1997) can be captured through an examination of their network architecture (Strogatz, 2001). Therefore, by characterising key features of the actual process architecture, the model reveals the otherwise hidden mechanisms through which information is transformed.

A related assumption is that as the scale of an engineering design project grows larger, the influence of each person in the process decreases, and simultaneously, the influence of the network architecture increases. This assumption is important because the model treats each person as a black box and captures only their information exchanges, their degree of participation in activities, and certain attributes such as their departmental affiliation. Consistent with the notion of emergence, the model assumes that there is an effect on the design process that goes beyond the linear sum of design outputs produced by each individual, an effect that lies in the particular network structure and composition in which each individual and activity are embedded. Based on this assumption, for design processes with complex architectures it makes sense to examine the network structure and composition, even if the model cannot capture each person's effect as an individual.

Levels of analysis

Grounded in social network analysis and graph theory (Wasserman and Faust, 1994, p. 17), the model assumes that the three selected levels of analysis—node, interface, and whole network—are sufficient to study the network's properties. Although the model does not explicitly cover additional levels, such as triads, they can be modelled as a subset of the whole network level and can be incorporated if required.

Process dynamics

For the presented model, I have assumed that a static representation is enough at the level of individual activities and interfaces. However, this assumption is flexible because no constraint in the model prevents activities and interfaces from being modelled in terms of their own dynamic evolution. The main challenge would be to acquire a dynamic benchmark or performance variable for each activity that would make such a dynamic analysis meaningful (considering the complexity that dynamic analysis adds to the model when applied as such a level of detail).

6.2 Limitations

Some of the model's main limitations are directly related to the current constraints of general network models of complex systems, including:

- The model does not allow incorporating agency or behaviour to each node, which makes the network only a representation of the actual process architecture through which information flows. On its own, this model does not simulate individual behaviours, and therefore, it is not suitable as a primarily predictive model that attempts to directly explains causal relationships. However, in combination with other approaches such as system dynamics and/or agent based modelling, the model can be enriched and used as an input for such purposes.
- The impact of heterogeneous components (various types of activities, people with different behaviours and functions, etc.) is partially captured through the measure of network composition; however, for simplicity, the potential effect of heterogeneity on network structure is not directly incorporated. That is to say, the model and the analytical methods presented here do not provide the means to examine the interplay between network structure and composition.

- In the analyses of the interplay of independent network variables with dependent performance measures, the proposed model does not differentiate between correlation and causation. Such differentiation must be implemented through a qualitative judgment or through further analyses outside the model's scope.

Method-specific limitations include:

- The information centrality metric and its nearness matrix (Stephenson and Zelen, 1989) are only defined for undirected networks; therefore, to compare the actual and the planned process architecture, we must first symmetrise the information exchanges between the organisations' members. Thus, when this metric is applied, the actual process architecture cannot capture the direction of information exchanges. However, as Morelli et al. (1995) found, at the aggregate level most information exchanges between project members are reciprocated (although the content changes), making the direction of information exchanges less relevant, especially in large projects.
- Ideally, each information exchange should be associated with a particular output-input exchange between identifiable activities. However, this association is not only hard to attain through already available information, but also hard to scale analytically through the proposed method. This difficulty stems from the network analysis method used to quantify the model, which assumes that all information exchanges are part of the same network structure in order to avoid the existence of multiplexity and the creation of multiple layers in the network structure (Kivelä et al., 2013). By keeping the analysis in the same layer, the method can integrate all information exchanges into a set of comparable information flows, which in turns allows the use of conventional statistical analysis and enables the comparison between actual and planned process architectures.
- The application of a 2-step clustering analysis and one-way ANOVA only provides a starting point for subsequent and more detailed quantitative and qualitative analyses to explore the link between the architectures of interfaces and activities and performance. This thesis suggests such analyses as opportunities for future research.
- The empirical analysis of the dynamic evolution of the actual process architecture presented in this thesis was constrained by limitations imposed by the nature of the dataset utilised. As a result, the evidence of patterns matching the expected evolution of GBC at each stage is not conclusive. This limitation was overcome in the paper submitted by the author to IEEE Transactions on Engineering Management entitled "Information Flow

through Stages of Complex Engineering Design Projects" (currently under review). The dataset used in that paper includes e-mail information exchanges, allowing for a fully dynamical analysis of both participation in activities and information exchanges. The results confirmed the findings presented in this thesis.

- Given the range of network sizes that may be expected for an analysis of this kind and the non-experimental settings, the proposed method cannot robustly control for the effect of different activity types. However, in large networks with variables approaching a normal distribution, the effect of different activity types could be identified and controlled. As a result, the proposed method relies on a qualitative judgement of the findings. The empirical experience obtained through the case studies demonstrated that providing this qualitative judgement was sufficiently simple for the company's personnel, and moreover, came as natural reaction during the assessment of the quantitative results.

6.3 Evaluation of outcomes

In what follows the success criteria set in section 3.2.3 are used to assess the degree to which the answers to the research questions fulfilled the defined research objectives.

6.3.1 Evaluation of outcomes for industry

The industrial success criteria for the outcomes of this research stated:

- The conceptual model should deliver an increased overview of the actual design process through a model that achieves face validity from the company's perspective. The model should be operationalisable, using information that is economically feasible to gather and representative of the process.
- The analytical methods should allow a quantitative characterisation of the actual design process architecture that provides companies with a practical and intuitive means to map their actual design processes and compare them with their design process plans.
- As a result of the framework's application, the case study participants should report increased awareness and improved understanding about their actual design process, the relationships between process architecture characteristics and process performance metrics, and the differences between their planned and actual process architectures.

Based on these success criteria and after a validation of the framework through interviews and presentations, participants in the two case studies found the framework was an improved way to describe their actual design processes and to increase their process overview. They reported that the framework achieved face validity in its capacity to provide a sufficiently accurate, albeit simplified, representation of reality.

The framework requires information that can be gathered through replicable and straightforward means, such as activity logs, internal communication platforms, and pre-existent process models, which makes it suitable for large-scale implementations. In addition, the information required to implement the analytical method is often readily available or easily acquired and can be built on bottom-up data traces rather than exclusively on top-down information or qualitative judgements. The outputs of the analytical method also are simple to interpret and visualise. The required software is already available, and this doctoral project also developed the foundations of a single, cloud-based software platform to automate and simplify the framework's implementation. This platform is available in its early form at **http://bit.ly/ESG-NetSights** (See appendix G for more details).

Concrete benefits of the framework for the two case studies

Exploratory case study: The results derived from the framework in the pilot case study (not included in this thesis) highlighted misalignments between the planned and the actual design processes, which were associated with lower performance levels for the most misaligned activities. Based on new insights that the framework provided, the case study decision makers decided to take the following actions:

- Revise description of roles and responsibilities to make them more explicit. This decision was triggered because the activities with lower than expected information flows were also the activities with performance problems. The insufficient information flow was associated to the mapping of people to activities; therefore, making the roles and responsibilities more explicit could reorganise and improve the information flows and increase their alignment with information dependencies.
- Schedule periodical coordination meetings to strengthen the weaker interfaces. Feedback from the framework's application facilitated the design of these coordination meetings and indicated the people who were required to attend.
- Allocate a permanent room for the new coordination meetings. An established permanent location for the meetings would make process visualisation aids, ideas, and other relevant information always available to support information flow between activities.

Descriptive case study: Through the interviews, the iterative development of design support, and the multiple process visualisations and analyses, company personnel were able to increase their awareness of their process, and in turn, to develop insights and new ideas to face their challenges. The two main issues that the company reported were limited overview of the actual process in relation to plans, and general 'interface problems'. The developed framework led to the following actions to improve the design process:

- Through the concrete conceptual model and visualisations, the company was able to make a clearer distinction between process, product, and organisational interfaces, and thus, could narrow issues more clearly.
- For subsequent projects, the company decided to create a new formal organisational role called 'interface lead'. The person fulfilling this role would be responsible for overall interface coordination at the system level and become a potentially active user of the NPr Framework. In addition, the company agreed to perform a more active, systematic mapping of interfaces through meetings and documentation that could be used to gather data in near real-time. The objective was to prevent identified interface problems and to proactively strengthen weak process interfaces before problems arise. The NPr Framework was reported as one of the influences for these new measures.
- Although the framework primarily focuses on mapping the process architecture in terms of activities and interfaces, it also can yield insights about key organisational members. The framework's analysis at the people level highlighted key information brokers who played an important and sometimes-unexpected role in the information flows between multiple activities. For example, shortly after the analysis, one such information broker left the organisation. As the analysis anticipated, the impact of his departure was also higher than what might have been expected without knowing the results that the framework provided.
- The framework can be applied to processes spanning multiple organisations. In fact, in this case external organisations were also identified as actors in the network (although for simplicity, the results presented here were narrowed to the focal company only). When the procurement network was added to the analysis, redundant information exchanges and lack of coordination between some internal functions and external organisations were revealed. Through the increased process overview, the company planned to improve the process to make the procurement process more efficient.

In addition to these specific industrial outcomes from the two case studies, the following more general outcomes can be expected when the NPr Framework is applied to other organisations designing engineering systems:

- **At the activity and interface levels** the framework enables a comparison of the architectures of all activities and interfaces and the identification of best practices in the specific company's context. For example, the company can ask: Is there a relationship between the number of people in an activity (or interface) and the performance of such activity (or interface)? If there is a relationship, what range is the optimal number of people? In relative terms, is the compositional diversity (in terms of departmental affiliations) of a given activity (or interface) high or low? Is it likely that the activity (or interface) problem is related to insufficient information exchanges between people?

  As the framework helps to answer these type of questions, it becomes a decision-support tool to redesign the organisation of individual activities and interfaces in greater detail than previously possible through alternative process architecture approaches. Also, although the same questions, data, and analysis apply to both activities and interfaces, best practices in architectural characteristics can differ between activities and interfaces. For instance, in a given process, a certain range of size, density, and compositional diversity of organisation networks might be associated with low activity performance and high interface performance. This difference is possible because large, diverse, and sparse activity architectures might be undesirable, but the same configuration might be ideal for a healthy interface between two activities with very different types of subsystems.

- **At the process level,** the framework enables more detailed tracking of the process because the actual and dynamic process patterns can be compared with the expected patterns at each system engineering stage. Similarly, the actual process architecture can be compared with the planned architecture based on information dependencies. These possible comparisons represent a new way for industry to prevent processes from exceeding time limitations and budgets through more active monitoring and intervention in the actual process architecture before expensive problems arise. In addition, because of the relationship between information flows and information dependencies, the actual process architecture can be a reference when eliciting information dependencies.

Based on all this evidence, I can claim that all industrial success criteria have been met.

6.3.2 Evaluation of outcomes for research

The academic success criteria for the outcomes of this research stated:

- The conceptual model should provide new insights for understanding the actual process architecture, compared with previously available approaches. The model should address the dynamic and multilevel nature of the design process architecture.
- The analytical method should provide a quantitative characterisation of the developed model. This characterisation should be comparable to planned design process models and integrate all levels of analysis defined by the conceptual model.
- The data-driven support should provide a flexible and quantitative platform for future research that seeks to identify relationships between the actual process architecture and process performance metrics.

Based on these success criteria, the proposed framework was indeed able to provide new insights for understanding the actual architecture of design processes. These insights are summarised as follows:

- The framework offers means to model activities and interfaces as organisation networks with their own characteristic structure and composition, while at the same time maintaining consistency throughout the three levels of analysis. A key advantage of the model is that it connects with already existent models of process and organisation architectures, particularly DSM-based models, expanding them at the intersection of process and organisation and allowing for comparable representations. The literature review revealed that no previous network-based models focused at the level of network structure and composition of individual activities and process interfaces.
- The framework provides a way to explore the evolution of actual process architecture through a modelling technique that does not depend on the technical sequence of tasks, but rather on actual information flows between activities. This feature allows a comparison of the model with design process stages as defined by more general, prescriptive models or by company milestones.
- Despite its limitations, the framework allows for a quantification of the model that is consistent at all three levels of analysis and computationally tractable. The framework permits the relationship between independent and dependent variables to be tested, and it

does not introduce significant artificial constraints to implementation of the conceptual model.

- The framework uses the information centrality nearness matrix (Stephenson and Zelen, 1989) to estimate information flows between activities, the Index of Qualitative Variation (Agresti and Agresti, 1977) to measure network compositional diversity, and two-step clustering techniques to facilitate further analysis and interpretation. The framework's application of these techniques is novel in the Engineering Design context, and their incorporation allows future research in this area to benefit from these analytical methods.

Based on this evidence, I maintain that all research success criteria have been met.

6.4 Comparison with other approaches

Although I have claimed that the NPr Framework developed in this thesis is a novel contribution to industry and research, I have of course drawn upon several features of previous network-based process models. In particular, the framework has benefited from the long tradition of analysing Engineering Design and R&D projects through the characterisation of their architectures. This tradition includes applied graph- and matrix-based approaches, such as those pioneered by scholars like Donald Steward and Thomas Allen (e.g. Steward 1981; Allen 1986). Their original contributions were expanded through a series of other studies, some of which were detailed in the literature review and also influenced this research.

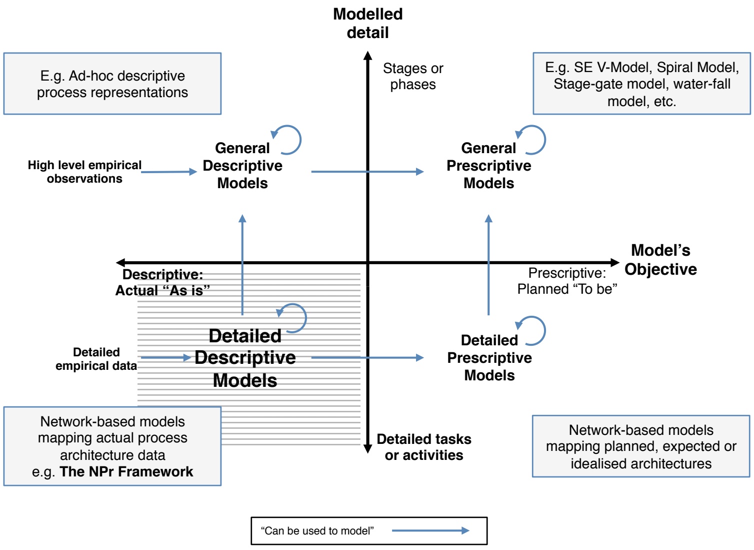


Figure 6-1: Overall position of the NPr Framework in the process model classification matrix

In the process model classification matrix previously introduced (section 2.1.4), the NPr Framework belongs to network-based process models that are detailed and descriptive, and are fed by actual process architecture data (figure 6-1, bottom left quadrant). However, other approaches and frameworks also occupy sections of this same space: approaches such as process architecture models that connect activities and tasks directly through estimations (section 2.3), approaches at the intersection of product and process architectures like the Signposting Framework (section 2.5.1), approaches at the intersection of process and organisation architectures like Sosa (2008), Durugbo et al. (2011), and Morelli et al. (1995) (section 2.5.2), and other cross-domain approaches like Yassine, Whitney, et al. (2003) and Maurer (2007) (section 2.5.3). Therefore, to refine the NPr Framework's position and facilitate its comparisons with other approaches, table 6-1 summarises the key features of the approaches closest to the NPr Framework (see also tables 2-4, 2-5, and 2-6) and highlights key differentiating elements. Some of the aspects that set the NPr Framework apart are its intersection of process and organisation architectures, the focus on actual information flows between activities, the multilevel, dynamic architectural characterisation that includes structure and composition, and the explicit means to test architecture-performance relationships.

Table 6-1: Summary of the NPr Framework features relevant to the posed research objectives in relation to other approaches

Approaches	Seeks to obtain		Allows		Features architecture				Method		
	Information flows, information dependencies or both	**Planned architecture, actual architecture or both**	**Comparison between planned and actual process architecture**	**Operational distinction between information dependencies and flows**	**Multilevel features**	**Dynamic**	**Structure**	**Composition**	**Valued direct and indirect paths**	**Identification of outliers**	**Architecture - performance test**
Intersection process-organisation with focus on organisation and/or alignment. **See table 2-5**	**Information flow** with focus on the organisation domain	**Both** with information flows focused on the organisation domain	No, but enables potential comparisons	Partial given the organisational focus	Some but focus on whole network architecture	Mainly static	Yes	Some compositional considerations	Mainly binary matrices as inputs. Indirect paths can be considered	Yes, mainly at the organisation level	Posed as a possibility
Intersection product-process with focus on process. **See table 2-5**	Actual **information dependencies** through interdependent component parameters	**Actual process architecture** based on **information dependencies**	No, but enables the quantification of actual information dependencies	Partial as it does not include the organisation	Some multilevel features	Some dynamic features	Yes	No	Can be valued and consider indirect paths	Yes	Posed as a possibility
Multi-domain intersection or domain independent. **See table 2-6**	Generic	Generic	It can be used but has not been formalised	No, but sets the basic elements to intersect and compare domains.	Allows for generic analysis at different levels of detail	Generic network level dynamics	Allows for detailed structural analysis with focus on whole network level	Some compositional considerations	Can be valued and consider indirect paths	Yes	Yes
Intersection process-organisation with focus on process: **The NPr Framework**	**Information flows**	**Actual process architecture** based on **information flows**	**Yes, explicitly**	**Yes, through a distinction between the process architecture based on activities and the process architecture based on tasks**	**Characterisation at the level of activities, interfaces and whole process**	**Yes, for the whole process network**	**Yes**	**Yes**	**Yes, all paths with fully valued matrices**	**Yes, at each level and based on clustering**	**Yes**

Feature assessment given the posed research objectives

Not valued	Low	Medium	High

6.5 Lessons learned

Throughout this research, I gained a number of valuable lessons, some of which have not been reported elsewhere in this thesis. This brief discussion of these lessons aims to facilitate the work of future researchers in this area.

6.5.1 Structured data sources

Engineering design projects leave more digital traces during the design process than most members of the organisation may be aware of. Moreover, even when the data is known to exist and is easy to gather, many organisations do not actively use the information to support decision making either because they do not have the required competencies and/or because they are unaware of the data's value for design process analysis (for a review of this topic see Sundararajan et al. [2013] and Provost and Fawcett [2013]). For this reason, it is important to actively identify valuable data sources and help companies discover them. Even when confidential data is involved, it often can be made anonymous without losing relevant information for process architecture. These data-sources should be identified early in the process, as otherwise information that might be already in existence could end up being manually gathered, wasting valuable time for the researcher and organisation. In addition, digital data traces are often far richer than expected due to its associated temporal metadata, allowing for dynamic analysis that otherwise would not be possible.

6.5.2 The advantage of focusing on activities

An analysis of organisation architecture in terms of specific people can be very relevant and useful to complement the analysis of the design process, but focusing on activities instead of people has interesting advantages that may make the difference between a feasible and an unfeasible study. The main issue when focusing on people are the sensitive political considerations inside organisations. For example activities can often be analysed without anonymity and can be associated with concrete performance measures. However this tend not to be the case for people, departments, or groups due to privacy or political reasons. This difference in perception generates an interesting space for research in the line of the NPr Framwork that utilises information about people and activities but focuses only on outputs at the activity, interface, or process level. In addition, people may come in and out of an organisation, but activities exist because of their real or expected value to the design process, which makes them more stable building blocks of the process architecture.

6.5.3 The importance of the right process view

The power of the correct visualisation should not be underestimated. The exploratory case study included a long time period in which I tested different process visualisations that often proved to be inadequate because of their complex (and sometimes complicated) layout. Moreover, when combined with the represented network's inherent complexity, many of those representation were not able to convey the right message. Combining information from different domains in the same visualisation unavoidably increases the representational challenges. As a result, whenever possible, the combination of multiple domain architectures should be avoided, and instead, a single-mode visualisation that contains all the relevant information should be preferred. Also, in the case of interactive visualisations, users ideally should be able to unfold the network and change between levels of analysis depending on their specific requirements.

6.6 Chapter summary

A good way to summarise the overall discussion and evaluation of the outcomes of this research is to review a figure introduced earlier on in this thesis (figure 4-2). The figure is re-introduced below as figure 6-2, showing now the shortened feedback path between the planned and actual architectures that the NPr Framework provides.

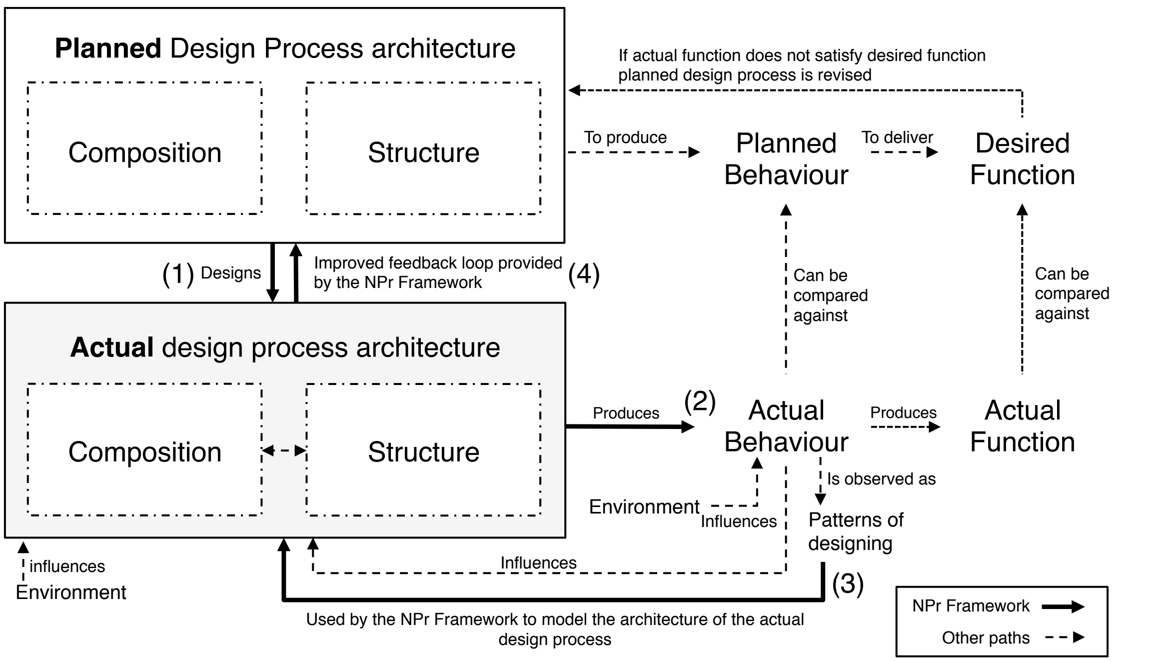


Figure 6-2: Revisiting the relationships between planned and actual architectures and the relationship between architecture, behaviour, and structure, using the NPr Framework.

Figure 6-2 shows the key elements that the the NPr Framework addresses and that were also the core elements of this research project, including:

- The relationship between architecture and performance (path from architecture to function); a set of methods were developed to test the architecture-performance relationship and empirical evidence was provided that such a relationship exists (at least in the examined case study).
- The division of architecture in structure and composition, providing a richer characterisation of the architecture than originally possible.
- The importance of direct feedback between the actual and planned process architectures, feedback that the framework enables through a quantitative characterisation of the actual process architecture and a shorter feedback loop to compare the planned and actual architecture (see new path following numbers 1 to 4 in figure 6-2).
- Mechanisms to use otherwise unused digital traces that reveal valuable patterns of designing (observed as behaviour during the process), and the means for structuring such patterns so that they can reveal the actual process architecture.

In addition, this chapter described key assumptions and limitations of this research project and the NPr Framework, a comparison of the NPr Framework with other approaches, and offered some insights about key lessons learned during the research project.

7 CONCLUSIONS

The paradox is only a conflict between reality and your feeling of what reality ought to be

—Richard Feynman

To characterise the actual design process architecture and to provide data-driven support for the design process of engineering systems, this thesis developed a framework composed of a conceptual model and analytical methods. Key and distinctive characteristics of the developed framework include its networked, multilevel, and dynamic nature, its emphasis on the actual process at the intersection of process and organisation architectures, its ability to test relationships between architecture and performance measures, and its capacity to compare the actual process architecture against the planned process. In addition, the framework can use digital traces produced throughout the process, instead of relying exclusively on the knowledge that experts in the organisation might provide.

The proposed conceptual model re-examines and enriches our understanding of design activities, process interfaces, and the whole process network, as well as the relationship between the actual and planned process, and the connections between the architectures of the process, product, and organisation domains. Although the model is aimed at the actual design process at the intersection of process and organisation, its inherent networked nature connects it with pre-existent models.

The analytical methods provide concrete means for the quantification of what otherwise would be a purely conceptual framework. The methods are a combination of applied graph theory, particularly approaches from social network analysis and statistical methods. The integration of the model and the methods during the framework's application to the design process of an engineering system provides data-driven support for evidence-based decision making to help redesign and improve the design process.

In this research, I sought to answer three research questions:

1. **How can we model the multilevel, dynamic, and actual design process architecture of engineering systems?**

   The answer this thesis provides to this question (chapter 4) is that the actual process architecture can be modelled as the intersection of two networks: a network of people exchanging information and a network of people performing activities. The bimodal network that emerges at this intersection can be dynamic, and its architecture can be modelled at three levels of analysis.

2. **How can we quantitatively characterise the model of the actual architecture so that it is analytically comparable to planned process architecture views of engineering systems?**

   The answer to the quantitative characterisation of the actual process architecture, detailed in chapter 4 and exemplified in chapter 5, is that both the structure and composition of the actual process architecture can be characterised using network-based metrics alongside metrics of compositional diversity. This combination creates transparent, replicable analytical methods.

3. **How can we connect a quantitative characterisation of the actual architecture with process performance metrics?**

   To answer this question, this thesis provides a set of simple steps to transform the otherwise de-contextualised characterisation into data-driven insights, achieved through a combination of visualisations, statistical techniques, and dynamic reference models based on standard design process stages (for the case of dynamic process architecture).

7.1 Research implications

The key research implications are:

- The architectures of design activities and process interfaces can now be characterised according to the structure and composition of their organisation network architectures, which improves our understanding of what usually has been treated as a black box of information transformation or information flow, black boxes that cannot be further characterised.
- At the activity and interface levels, case study evidence is provided about relationships between the architecture of activities and interfaces, identifying a set of basic network characteristics that can be used and expanded in future studies.
- At the whole process level, empirical evidence based on large datasets is provided about the relationships between the proposed measures for information centrality, information centralisation, and design stages. These relationships quantify information network

properties for different stages of the design process, enriching previous descriptions and interpretations of the stages and allowing design researchers to develop process models that better fit observed project patterns.

- The framework allows rich digital data traces of the process and organisation domains to be combined to build a bottom-up model of the actual process architecture, which enables the scalable analysis of large projects. As a result, model makers need not rely exclusively on the subjective and limited views of project participants and can avoid confusing or mixing process plans, expectations, and estimations with actual processes.
- As part of the framework's development, a set of Excel macros and a web-based tool called 'Net-Sights' was also developed to facilitate some aspects of the network analysis the framework requires. These tools are available for researchers and practitioners at **http://bit.ly/ESG-NetSights.**

7.2 Managerial implications

Managerial implications include new data-driven support to facilitate the work and decision-making processes of interface and project managers and to detect process anomalies. As an example, the application of the developed framework in the descriptive case study resulted in an improved overview of the interfaces, raised awareness about the importance of the actual process architecture, contributed to the creation of a new job position, and supported new initiatives to map and actively manage activities and process interfaces.

When the analysis is applied dynamically, the framework also can be used to highlight periods in the process in which multiple areas concurrently increase their information centrality, potentially draining resources and generating complex coordination scenarios. Knowing more about these periods can help to defer activities that do not need to be concurrently active, while prioritising the ones with coupled subsystems that do require concurrency or iterations.

In addition, with appropriate tools to structure and analyse existent information, the framework facilitates the early identification of unexpected or undesirable information flow patterns, by comparing the project's evolving stages with the actual process workflow. Such a comparison between idealised stages and the actual process allows an assessment of the project's progress, aids in the prescription of changes, and allows managers to monitor the project, enabling an improved process overview.

7.3 Directions for future research

Further studies can continue the development of this framework through the inclusion of more and especially tailored network and non-network metrics to deliver a more comprehensive characterisation of the architectures at each level of analysis. Also, additional cases from different industries and contexts would help to identify if a set of common architectural characteristics consistently links with certain performance outputs. If a consistent connection were found between the actual process architecture and performance, the impact would be profound and would significantly affect future prescriptive models of the design process and support tools.

The positive results of the developed framework open the door for additional implementations, which could be based on a more automated data-gathering process, exclusively using digital data-traces as information sources. Such a deployment of the framework requires further development of the Net-Sights platform, but the advantages of an on-going analysis of the process-architecture that requires minimal maintenance efforts could well justify such development. Data sources that could be mined to automate the framework's implementation may include e-mail communication, data logs of events, document databases, event logging systems, and other process related datasets. A permanent implementation of the framework to map the actual process architecture has the additional advantage of gathering large datasets from multiple projects. Such rich data would allow us to gain predictive power to anticipate process problems and proactively suggest changes to the process architecture. This power also would allow implementation of machine learning algorithms to identify in real time the architectural patterns linked with particularly high or low performance. The first steps in the direction of this suggested future research have been already taken in the paper 'Information Flow through Stages of Complex Engineering Design Projects: A Dynamic Network Analysis Approach' (currently under review with IEEE TEM). In that paper, all information exchanges and participation in activities relied exclusively on digital data traces. The results are consistent with the ones presented in this thesis.

In terms of statistical analyses and other tests to determine the relationship between the architecture of activities, interfaces, the whole process and performance, this thesis opens interesting opportunities for additional studies. They include gathering activity and interface level data on complexity and interdependence, which would allow detailed regression analyses where the effect of different architectural characteristics can be estimated and key features of activities and interfaces can be statistically controlled for.

By combining the proposed framework with other approaches such as Process Mining (van der Aalst, 2011; van der Aalst et al., 2003), System Dynamics (Sterman, 2007), multi-agent models of the design process (Alexiou, 2007), and semantic analysis of the rich content produced during the process, we could acquire a more detailed and accurate characterisation of the process without significantly increasing data-acquisition costs. Such detail is possible because the combination of these types of approaches can utilise a common pool of big-data digital traces, combined under the umbrella of the ever-growing computational social science field.

With the complex cross-domain networks that can be explored through the NPr Framework, the challenge of appropriately visualising the computational outputs increases. Future research is planned to address this challenge and test the most effective and efficient means for cross-domain network visualisation. Future research will include the development and testing of interactive platforms to facilitate the use of the rich information produced.

The proposed framework can also be applied in non-design applications, as it is flexible enough to map different kinds of processes. For example, the framework may be used in operations and manufacturing, where tight control is required between the actual and the planned process. Finally, it can also be applied in inter-organisational and industry projects to map collaborative potential, where understanding complex knowledge and technology landscapes is considered critical to foster true open innovation.

REFERENCES

Van der Aalst, W.M.P. (2011), *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer, Berlin, doi:10.1007/978-3-642-19345-3.

Van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G. and Weijters, A.J.M.M. (2003), "Workflow mining: A survey of issues and approaches", *Data & Knowledge Engineering*, Vol. 47 No. 2, pp. 237–267.

Abbott, M. (2013), *Understanding and applying research design*, Wiley, Hoboken.

Abraham, A. (2010), *Computational social network analysis trends, tools and research advances*, Springer, New York.

Agresti, A. and Agresti, B.F. (1977), "Statistical Analysis of Qualitative Variation", in Schuessler, K.F. (Ed.),*Sociological Methodology*, Jossey-Bass, San Francisco, pp. 204–237.

Albert, R. and Barabási, A.-L. (2002), "Statistical mechanics of complex networks", *Reviews of modern physics*, APS, Vol. 74 No. 1, p. 47.

Alderson, D.L. (2008), "Catching the 'Network Science' Bug: Insight and Opportunity for the Operations Researcher", *Operations Research*, Vol. 56 No. 5, pp. 1047–1065.

Alexiou, K. (2007), *Understanding Multi-Agent Design as Coordination*, PhD Thesis, University College London.

Alexiou, K. (2010), "Coordination and emergence in design", *CoDesign*, Vol. 6 No. 2, pp. 75–97.

Alexiou, K., Zamenopoulos, T., Johnson, J.H. and Gilbert, S.J. (2009), "Exploring the neurological basis of design cognition using brain imaging: some preliminary results", *Design Studies*, Elsevier Ltd, Vol. 30 No. 6, pp. 623–647.

Allen, J., James, A.D. and Gamlen, P. (2007), "Formal versus informal knowledge networks in R&D: a case study using social network analysis", *R&D Management*, Vol. 37 No. 3, pp. 179–196.

Allen, P., Maguire, S. and McKelvey, B. (Eds.). (2011), *The SAGE handbook of complexity and management*, SAGE, London.

Allen, T. (1986), "Organizational structure, information technology and R&D productivity", *IEEE Transactions on Engineering Management*, Vol. EM-33 No. 4, pp. 212–217.

Allen, T.J. and Henn, G. (2006), *The Organization and Architecture of Innovation: Managing the Flow of Technology*, Butterworth-Heinemann, San Francisco.

Anderson, B.S., Butts, C. and Carley, K.M. (1999), "The interaction of size and density with graph-level indices", *Social Networks*, Vol. 21 No. 3, pp. 239–267.

Austin, S., Steele, J., Macmillan, S., Kirby, P. and Spence, R. (2001), "Mapping the conceptual design activity of interdisciplinary teams", *Design Studies*, Vol. 22 No. 3, pp. 211–232.

Baldwin, C., MacCormack, A. and Rusnak, J. (2013), *Hidden Structure: Using Network Methods to Map System Architecture* (No. 13-093), Harvard Business School.

Barabási, A. (2002), *Linked: The New Science of Everything*, Perseus Publishing, Cambridge.

Barabási, A. (2012), *Network Science*, (Hövel, P.,Ed.), BarabasiLab, Boston.

Bartolomei, J.E., Hastings, D.E., de Neufvill, R. and Rhodes, D.H. (2012), "Engineering Systems Multiple-Domain Matrix: An organizing framework for modeling large-scale complex systems", *Systems Engineering*, Vol. 15 No. 1, pp. 41–61.

Bastian, M., Heymann, S. and Jacomy, M. (2009), "Gephi: an open source software for exploring and manipulating networks", *International AAAI Conference on Weblogs and Social Media*.

Batallas, D.A. and Yassine, A.A. (2006), "Information Leaders in Product Development Organizational Networks: Social Network Analysis of the Design Structure Matrix", *IEEE Transactions on Engineering Management*, Vol. 53 No. 4, pp. 570–582.

Becker, G. (1992), "The Division of Labor, Coordination Costs, and Knowledge", *The Quarterly Journal of Economics*, Vol. 107 No. 4, pp. 1137 – 1160.

Bedny, G. and Karwowski, W. (2004), "Activity theory as a basis for the study of work", *Ergonomics*, Vol. 47 No. December 2014, pp. 134–153.

Behrend, F.D. and Erwee, R. (2009), "Mapping knowledge flows in virtual teams with SNA", *Journal of Knowledge Management*, Vol. 13 No. 4, pp. 99–114.

Bender, B. (2003), "Task design and task analysis for empirical studies into design activity", *Journal of Engineering Design*, Vol. 14 No. 4, pp. 399–408.

Blackler, F. (1993), "Knowledge and the theory of organizations: Organizations as activity systems and the reframing of management", *Journal of Management Studies*, Vol. 30 No. 6, pp. 863–884.

Blessing, L. (1994), *A process-based approach to computer supported engineering design*, University of Twente.

Blessing, L. and Chakrabarti, A. (2009), *DRM, a Design Research Methodology*, Springer, London.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. and Hwang, D. (2006), "Complex networks: Structure and dynamics", *Physics Reports*, Vol. 424 No. 4-5, pp. 175–308.

Boehm, B. (2000), *Spiral Development: Experience, Principles, and Refinements*, Carnegie Mellon.

Bonacich, P. (1987), "Power and Centrality: A Family of Measures", *The American Journal of Sociology*, Vol. 92 No. 5, pp. 1170–1182.

Borgatti, S. (2005), "Centrality and network flow", *Social Networks*, Vol. 27 No. 1, pp. 55–71.

Borgatti, S., Carley, K.M. and Krackhardt, D. (2006), "On the robustness of centrality measures under conditions of imperfect data", *Social Networks*, Vol. 28, pp. 124–136.

Borgatti, S. and Everett, M.G. (2006), "A Graph-theoretic perspective on centrality", *Social Networks*, Vol. 28 No. 4, pp. 466–484.

Borgatti, S., Everett, M.G. and Freeman, L.C. (2002), "Ucinet for Windows: Software for Social Network Analysis", Analytic Technologies.

Borgatti, S., Everett, M.G. and Johnson, J.C. (2013), *Analyzing Social Networks*, SAGE Publications, London.

Borgatti, S. and Foster, P. (2003), "The Network Paradigm in Organizational Research: A Review and Typology", *Journal of Management*, Vol. 29 No. 6, pp. 991–1013.

Borgatti, S. and Halgin, D.S. (2011), "On Network Theory", *Organization Science*, Vol. 22 No. 5, pp. 1168–1181.

Boujut, J.-F. and Laureillard, P. (2002), "A co-operation framework for product–process integration in engineering design", *Design Studies*, Vol. 23 No. 6, pp. 497–513.

Braha, D. (2006), "The Structure and Dynamics of Complex Product Design", in Braha, D., Minai, A.A. and Bar-Yam, Y. (Eds.),*Complex Engineered Systems: Science Meets Technology*, Springer, Berlin, pp. 40–71.

Braha, D. and Bar-Yam, Y. (2004), "Information flow structure in large-scale product development organizational networks", *Journal of Information Technology*, Vol. 19 No. 4, pp. 244–253.

Braha, D. and Bar-Yam, Y. (2007), "The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results", *Management Science*, Vol. 53 No. 7, pp. 1127–1145.

Browning, T.R. (2001), "Applying the design structure matrix to system decomposition and integration problems: a review and new directions", *IEEE Transactions on Engineering Management*, Vol. 48 No. 3, pp. 292–306.

Browning, T.R. (2002), "Process integration using the design structure matrix", *Systems Engineering*, Vol. 5 No. 3, pp. 180–193.

Browning, T.R. (2009), "The Many Views of a Process: Toward a Process Architecture Framework for Product Development Processes", *Systems Engineering*, Vol. 12 No. 1, pp. 69–90.

Browning, T.R. and Eppinger, S.D. (2002), "Modeling impacts of process architecture on cost and schedule risk in product development", *IEEE Transactions on Engineering Management*, Vol. 49 No. 4, pp. 428–442.

Browning, T.R., Fricke, E. and Negele, H. (2006), "Key concepts in modeling product development processes", *Systems Engineering*, Vol. 9 No. 2, pp. 104–128.

Browning, T.R. and Ramasesh, R. V. (2007), "A Survey of Activity Network-Based Process Models for Managing Product Development Projects", *Production and Operations Management*, Vol. 16 No. 2, pp. 217–240.

Bruce, M. and Cooper, R.G. (2000), *Creative product design: a practical guide to requirements capture management*, Wiley, New York.

Bruun, H.P.L., Mortensen, N.H. and Harlou, U. (2013), "Interface diagram: Design tool for supporting the development of modularity in complex product systems", *Concurrent Engineering: Research and Applications*, Vol. 22 No. 1, pp. 62–76.

Bucciarelli, L.L. (1984), “Reflective Practice in Engineering Design”, *Design Studies*, Vol. 5 No. 3, pp. 185–190.

Bucciarelli, L.L. (1988), “An ethnographic perspective on engineering design”, *Design Studies*, Vol. 9 No. 3, pp. 159–168.

Buede, D. (2009), *The Engineering Design of Systems: models and methods*, Wiley, Hoboken.

Burt, R.S. (1992), “The social structure of competition”, in Nohria, N. and Eccles, R.G. (Eds.),*Networks and organizations: Structure, form, and action*, Harvard Business School Press, Boston, pp. 57–91.

Busby, J.S. (1998), “The neglect of feedback in engineering design organisations”, *Design Studies*, Vol. 28 No. 1977, pp. 103–117.

Calvano, C.N. and John, P. (2004), “Systems engineering in an age of complexity”, *Systems Engineering*, Vol. 7 No. 1, pp. 25–34.

Campos Silva, D.D., Santiago, L.P. and Silva, P.M.S. (2012), “Impact of Premature Information Transfer on Cost and Development Time of Projects”, *IEEE Transactions on Engineering Management*, Vol. 59 No. 4, pp. 692–704.

Cardella, M.E., Atman, C.J. and Adams, R.S. (2006), “Mapping between design activities and external representations for engineering student designers”, *Design Studies*, Vol. 27 No. 1, pp. 5–24.

Carpenter, M.A., Li, M. and Jiang, H. (2012), “Social Network Research in Organizational Contexts A Systematic Review of Methodological Issues and Choices”, *Journal of Management*, Vol. 38 No. 4, pp. 1328–1361.

Carrington, P.J., Scott, J. and Wasserman, S. (2005), *Models and Methods in Social Network Analysis*, (Carrington, P.J., Scott, J. and Wasserman, S.,Eds.), Cambridge University Press, Cambridge.

Cash, P., Stanković, T. and Štorga, M. (2014), “Using visual information analysis to explore complex patterns in the activity of designers”, *Design Studies*, Vol. 35 No. 1, pp. 1–28.

Cataldo, M., Herbsleb, J.D. and Carley, K.M. (2008), “Socio-Technical Congruence: A Framework for Assessing the Impact of Technical and Work Dependencies on Software Development Productivity”, *Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement - ESEM ’08*, ACM Press, New York, New York, USA, p. 2.

Chen, L. and Gable, G.G. (2013), “Larger or Broader: Performance Implications of Size and Diversity of the Knowledge Worker’s Egocentric Network.”, *Management and Organization Review*, Vol. 9 No. 1, pp. 139–165.

Chira, C. (2005), *The Development of a Multi-Agent Design Information Management and Support System*, PhD Thesis, Galway-Mayo Institute of Technology.

Christian, A.D. (1995), *Simulation of information flow in design*, PhD thesis, Massachusetts Institute of Technology.

Cilliers, P. (2000), *Complexity and postmodernism: understanding complex systems*, Routledge, London.

Clark, J.O. (2008), “System of systems engineering and family of systems engineering from a standards perspective", *2008 IEEE International Conference on System of Systems Engineering*, IEEE, doi:10.1109/SYSOSE.2008.4724201.

Clarkson, P.J. and Eckert, C.M. (2005), *Design process improvement: A review of current practice*, (Clarkson, J. and Eckert, C.,Eds.), Springer London, London, doi:10.1007/978-1-84628-061-0.

Clarkson, P.J. and Hamilton, J.R. (2000), "'Signposting', A Parameter-driven Task-based Model of the Design Process", *Research in Engineering Design*, Vol. 12, pp. 18–38.

Colfer, L. and Baldwin, C. (2010), "The mirroring hypothesis: Theory, evidence and exceptions", *Harvard Business School Finance Working Paper No. 10-058*, Cambridge, MA.

Collins, S.T., Bradley, J. and Yassine, A.A. (2010), "Analyzing Product Development Task Networks to Examine Organizational Change", *IEEE Transactions on Engineering Management*, Vol. 57 No. 3, pp. 513–525.

Collins, S.T., Yassine, A.A. and Borgatti, S. (2009), "Evaluating Product Development Systems Using Network Analysis", *Systems Engineering*, Vol. 12 No. 1, pp. 55–68.

Cooper, R.G. (1990), "Stage-Gate Systems: A New Tool for Managing New Products", *Business Horizons*, No. May-June, pp. 44–54.

Cooper, R.G. (1994), "Third-Generation New Product Processes", *Journal of Product Innovation Management*, Vol. 11 No. 1, pp. 3–14.

Crawley, E., de Weck, O., Eppinger, S.D., Magee, C.L., Moses, J., Seering, W., Schindall, J., et al. (2004), *The Influence of Architecture in Engineering Systems*, Cambridge: Massachusetts Institute of Technology.

Cross, N. (2000), *Engineering Design Methods: Strategies for Product Design*, John Wiley & Sons, Chichester, Third., Vol. 58.

Cross, N. (2004), "Expertise in design: an overview", *Design Studies*, Vol. 25 No. 5, pp. 427–441.

Cross, N. and Cross, A.C. (1995), "Observations of teamwork and social processes in design", *Design Studies*, Vol. 16, pp. 143–170.

Culley, S.J. (2014), "Revisiting Design as an Information Processing Activity", in Chakrabarti, A. and Blessing, L.T.M. (Eds.),*An Anthology of Theories and Models of Design*, Springer London, London, pp. 371–394.

D'Agostino, G. and Scala, A. (Eds.). (2014), *Networks of Networks: The Last Frontier of Complexity*, Springer, London, doi:10.1007/978-3-319-03518-5.

Danilovic, M. and Browning, T.R. (2007), "Managing complex product development projects with design structure matrices and domain mapping matrices", *International Journal of Project Management*, Vol. 25 No. 3, pp. 300–314.

Davidsen, P. (1992), "The Structure-Behavior Diagram: Understanding the relationship between structure and behavior in complex dynamic systems", *Proceedings of System Dynamics Conference*, The Systems Dynamics Society, Utrecht, pp. 127–136.

Dorst, K. and Vermaas, P.E. (2005), "John Gero's function-behaviour-structure model of designing: A critical analysis", *Research in Engineering Design*, Vol. 16, pp. 17–26.

Dougherty, D. (2008), "Bridging Social Constraint and Social Action to Design Organizations for Innovation", *Organization Studies*, Vol. 29 No. 3, pp. 415–434.

Drucker, P.F. (1998), "The Discipline of Innovation", *Harvard Business Review*.

Durugbo, C. (2015), "Modelling information for collaborative networks", *Production Planning & Control*, Taylor & Francis, Vol. In press, doi:10.1080/09537287.2013.847217.

Durugbo, C., Hutabarat, W., Tiwari, A. and Alcock, J.R. (2011), "Modelling collaboration using complex networks", *Information Sciences*, Elsevier Inc., Vol. 181 No. 15, pp. 3143–3161.

Easley, D. and Kleinberg, J. (2010), *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, Cambridge.

Eckert, C.M. (2001), "The Communication Bottleneck in Knitwear Design: Analysis and Computing Solutions", *Computer Supported Cooperative Work (CSCW)*, Vol. 10 No. 1, pp. 29–74.

Eckert, C.M., Maier, A.M. and McMahon, C. (2005), "Communication in Design", in Clarkson, J. and Eckert, C. (Eds.),*Design Process Improvement: A review of current practice*, Springer, Cambridge, p. 560.

Eckert, C.M. and Stacey, M. (2001), "Dimensions of Communication in Design", *Proceedings of the 13th International Conference on Engineering Design (ICED 2001)*, Glasgow, pp. 473–480.

Eisenhardt, K.M. and Graebner, M.E. (2007), "Theory building from cases: Opportunities and challenges", *Academy of Management Journal*, Vol. 50 No. 1, pp. 25–32.

Engeström, Y., Miettinen, R. and Punamäki-Gitai, R.-L. (Eds.). (1992), *Perspectives on activity theory*, Cambridge University Press, Cambridge.

Eppinger, S.D. (2001), "Innovation at the Speed of Information", *Harvard Business Review*, No. January, pp. 149–158.

Eppinger, S.D. and Browning, T.R. (2012), *Design Structure Matrix Methods and Applications*, The MIT Press, Cambridge.

Eppinger, S.D. and Salminen, V. (2001), "Patterns of product development interactions", *Proceedings of the 13th International Conference on Engineering Design (ICED 2001)*, Glasgow.

Eppinger, S.D., Whitney, D.E., Smith, R.P. and Gebala, D.A. (1994), "A Model-Based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, Vol. 6, pp. 1–13.

Estrada, E., Fox, M., Higham, D.J. and Oppo, G.-L. (Eds.). (2010), *Network Science: Complexity in Nature and Technology*, *Science*, Springer, London, Vol. 345, doi:10.1007/978-1-84996-396-1.

Evans, S. and Jukes, S. (2000), "Improving co-development through process alignment", *International Journal of Operations & Production Management*, Vol. 20 No. 8, pp. 979–988.

Faust, K. (2006), "Comparing Social Networks: Size, Density, and Local Structure", *Metodološki zvezki*, Vol. 3 No. 2, pp. 185–216.

Federal Aviation Administration. (2014), *Boeing 787–8 Critical Systems Review Team Report*, CSRT.

Felekoglu, B., Maier, A.M. and Moultrie, J. (2013), "Interactions in new product development: How the nature of the NPD process influences interaction between teams and management", *Journal of Engineering and Technology Management*, Elsevier B.V., Vol. 30 No. 4, pp. 384–401.

Fixson, S.K. (2007), "Modularity and Commonality Research: Past Developments and Future Opportunities", *Concurrent Engineering*, Vol. 15 No. 2, pp. 85–111.

Fortunato, S. (2010), "Community detection in graphs", *Physics Reports*, Physics and Society; Statistical Mechanics; Information Retrieval; Biological Physics; Computational Physics; Quantitative Methods, , No. 486, pp. 75–174.

Frankfort-Nachmias, C. and Leon-Guerrero, A. (2011), "Measures of Variability", *Social Statistics for a Diverse Society*, Pine Forge Press, pp. 135–176.

Freeman, L.C. (1977), "A Set of Measures of Centrality Based on Betweenness", *Sociometry*, Vol. 40 No. 1, p. 35.

Freeman, L.C. (1979), "Centrality in social networks conceptual clarification", *Social networks*, Vol. 1 No. 1968, pp. 215–239.

Gemünden, H.G., Salomo, S. and Hölzle, K. (2007), "Role Models for Radical Innovations in Times of Open Innovation", *Creativity and Innovation Management*, Vol. 16 No. 4, pp. 408–421.

Gericke, K. and Blessing, L. (2012), "An analysis of design process models across disciplines", *Proceedings of DESIGN 2012, the 12th International Design Conference*, Dubrovnik, pp. 171–180.

Gero, J.S. and Kannengiesser, U. (2002), "The situated function – behaviour – framework", in Gero, J. (Ed.),*Artificial Intelligence in Design*, Kluwer Academic Publishers, Dordrecht, pp. 89–104.

Gero, J.S. and Kannengiesser, U. (2004), "The situated function-behaviour-structure framework", *Design Studies*, Vol. 25 No. 2004, pp. 373–391.

Gero, J.S. and Kannengiesser, U. (2007), "A function–behavior–structure ontology of processes", *AI EDAM: Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Vol. 21, pp. 379–391.

Giles, J. (2012), "Computational social science: Making the links.", *Nature*, Vol. 488 No. 7412, pp. 448–50.

Gloor, P. (2013), "Condor", Galaxy Advisors, Cambridge, Massachusetts.

Gloor, P. and Zhao, Y. (2004), "TeCFlow – A Temporal Communication Flow Visualizer for Social Network Analysis", *ACM CSCW Workshop on Social Networks*, Chicago.

Gokpinar, B., Hopp, W. and Iravani, S. (2010), "The Impact of Misalignment of Organizational Structure and Product Architecture on Quality in Complex Product Development", *Management Science*, Vol. 56 No. 3, pp. 468–484.

Gordon, D.M. (2014), "The ecology of collective behavior", *PLoS biology*, Vol. 12 No. 3, doi:10.1371/journal.pbio.1001805.

Gould, V. and Fernandez, R.M. (1989), "Structures of Mediation: A Formal Approach to Brokerage in Transaction Networks", *Sociological methodology*, Vol. 19, pp. 89–126.

Gupta, A.K., Raj, S.P. and Wilemon, D. (1985), "The R&D-marketing interface in high-technology firms", *Journal of Product Innovation Management*, Elsevier, Vol. 2 No. 1, pp. 12–24.

Hales, C. and Gooch, S. (2004), *Managing Engineering Design*, Springer, London, doi:10.1007/978-0-85729-394-7.

Halgin, D.S. and Borgatti, S. (2012), *An Introduction to Personal Network Analysis and Tie Churn Statistics using E-NET*, LINKS Center for Social Network Analysis, University of Kentucky, Kentucky, KY.

Hanneman, R.A. and Riddle, M. (2005), *Introduction to social network methods*, University of California, Riverside.

Haskins, C., Forsberg, K., Krueger, M., Walden, D. and Hamelin, R.D. (2011), *Systems Engineering Handbook*, (Cecilia Haskins,Ed.), International Council on Systems Engineering.

Heisig, P., Clarkson, P.J. and Vajna, S. (Eds.). (2010), *Modelling and Management of Engineering Processes*, Springer, London, doi:10.1007/978-1-84996-199-8.

Hicks, B. (2013), "The language of collaborative engineering projects", *Proceedings of the 19th International Conference on Engineering Design (ICED 2013)*, Seoul.

Holland, J.H. (1997), "Emergence", *Philosophica*, Vol. 59 No. 1, pp. 11–40.

Holme, P. and Saramäki, J. (2012), "Temporal networks", *Physics Reports*, Vol. 519 No. 3, pp. 97–125.

Hossain, L. (2009), "Effect of organisational position and network centrality on project coordination", *International Journal of Project Management*, Elsevier Ltd and IPMA, Vol. 27 No. 7, pp. 680–689.

Hossain, L., Murshed, S.T. and Uddin, S. (2013), "Communication network dynamics during organizational crisis", *Journal of Informetrics*, Vol. 7 No. 1, pp. 16–35.

Howard, T.J., Culley, S.J. and Dekoninck, E. (2008), "Describing the creative design process by the integration of engineering design and cognitive psychology literature", *Design Studies*, Vol. 29 No. 2, pp. 160–180.

Hubka, V. (1982), *Principles of engineering design*, Butterworth Scientific Press, Guildford.

Hubka, V. (1996), *Design science: introduction to needs, scope and organization of engineering design knowledge*, Springer, New York.

Hubka, V., Andreasen, M.M. and Eder, W.E. (1988), *Practical Studies in Systematic Design*, Butterworth-Heinemann, London.

IBM Corp. (2001), *SPSS two-step cluster analysis*.

IBM Corp. (2011), "IBM SPSS Statistics for Windows", Armonk, NY.

IEEE Standards Board. (1989), *IEEE Standard Glossary of Modeling and Simulation Terminology - Std 610.3-1989*, IEEE.

IEEE Standards Board. (2000), *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems - Std 1471-2000*, IEEE.

Iordache, O. (2011), *Modeling multi-level systems*, Springer, Berlin, Vol. 41, doi:10.1126/science.1247727.

Janis, I.L. (1982), *Groupthink: psychological studies of policy decisions and fiascoes*, Houghton Miffli, Boston.

Jansen, J.J.P., Van Den Bosch, F.A.J. and Volberda, H.W. (2006), "Exploratory Innovation, Exploitative Innovation, and Performance: Effects of Organizational Antecedents and Environmental Moderators", *Management Science*, Vol. 52 No. 11, pp. 1661–1674.

Jepsen, L.B. (2013), "Complex New Product Development Projects: How the Project Manager's Information Sharing With Core Actors Changes Over Time", *Project Management Journal*, Vol. 44 No. 6, pp. 20–35.

Jerome, L.W. (2012), "Innovation in social networks: knowledge spillover is not enough", *Knowledge Management Research & Practice*, Nature Publishing Group, Vol. 11 No. 4, pp. 422–431.

Johnson, J.D. (1992), "Approaches to organizational communication structure", *Journal of Business Research*, Vol. 25 No. 2, pp. 99–113.

Johnson, N. (2007), *Two's company, three is complexity: a simple guide to the science of all sciences*, Oneworld, Oxford.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y. and Porter, M.A. (2013), *Multilayer Networks* (Physics and Society), *arXiv*, eprint arXiv:1309.7233, arXiv.

Kleinsmann, M., Valkenburg, R. and Buijs, J. (2007), "Why do(n't) actors in collaborative design understand each other? An empirical study towards a better understanding of collaborative design", *CoDesign*, Vol. 3 No. 1, pp. 59–73.

Kossiakoff, A., Sweet, W.N. and Seymour, S. (2011), *Systems Engineering Principles and Practice*, John Wiley & Sons, Hoboken.

Kratzer, J., Gemünden, H.G. and Lettl, C. (2008), "Revealing dynamics and consequences of fit and misfit between formal and informal networks in multi-institutional product development collaborations", *Research Policy*, Vol. 37 No. 8, pp. 1356–1370.

Kreimeyer, M. and Lindemann, U. (2011), *Complexity Metrics in Engineering Design: Managing the Structure of Design Processes*, Springer, London, doi:10.1007/978-3-642-20963-5.

Kruskal, J. and Wish, M. (1978), *Multidimensional Scaling*, Sage Publications, Beverly Hills.

Labianca, G. (2004), "Group Social Capital and Group Effectiveness: the Role of Informal Socializing Ties.", *Academy of Management Journal*, Vol. 47 No. 6, pp. 860–875.

Lakemond, N., Johansson, G., Magnusson, T. and Säfsten, K. (2007), "Interfaces between technology development, product development and production: Critical factors and a conceptual model", *International Journal of Technology Intelligence and Planning*, Vol. 3 No. 4, pp. 317–330.

Leenders, R.T.A.J., Kratzer, J. and Engelen, J.M.L. Van. (2007), "Innovation team networks: the centrality of innovativeness and efficiency", *International Journal of Networking and Virtual Organisations*, Vol. 4 No. 4, p. 459.

Lindemann, U. (2003), *Human behaviour in design: individuals, teams, tools*, Springer, New York.

Lindemann, U., Maurer, M.S. and Braun, T. (2009), *Structural Complexity Management*, Springer, Berlin.

Loch, C. and Kavadias, S. (2008), *Handbook of New Product Development Management*, (Loch, C. and Kavadias, S.,Eds.), Butterworth-Heinemann.

MacCormack, A., Baldwin, C. and Rusnak, J. (2012), "Exploring the duality between product and organizational architectures: A test of the 'mirroring' hypothesis", *Research Policy*, Elsevier B.V., Vol. 41 No. 8, pp. 1309–1324.

Madni, A.M. and Sievers, M. (2014), "Systems Integration: Key Perspectives, Experiences, and Challenges", *Systems Engineering*, Vol. 17 No. 1, pp. 37–51.

Magee, C.L. and de Weck, O. (2004), "Complex System Classification", *Fourteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE)*, International Council On Systems Engineering (INCOSE).

Magurran, A.E. (1988), *Ecological Diversity and Its Measurement*, Springer, Dordrecht, doi:10.1007/978-94-015-7358-0.

Maier, A.M., Eckert, C.M. and Clarkson, P.J. (2005), "A meta-model for communication in engineering design", *CoDesign*, Vol. 1 No. 4, pp. 243–254.

Maier, A.M., Eckert, C.M. and Clarkson, P.J. (2006), "Identifying requirements for communication support: A maturity grid-inspired approach", *Expert Systems with Applications*, Vol. 31 No. 4, pp. 663–672.

Maier, A.M., Eckert, C.M. and Clarkson, P.J. (2009), "Towards managing team interfaces: An elicitation of factors influencing communication", *Proceedings of the 17th International Conference on Engineering Design (ICED 2009)*, Stanford, pp. 275–286.

Maier, A.M., Kreimeyer, M., Hepperle, C., Eckert, C.M., Lindemann, U. and Clarkson, P.J. (2008), "Exploration of Correlations between Factors Influencing Communication in Complex Product Development", *Concurrent Engineering*, Vol. 16 No. 1, pp. 37–59.

Maier, A.M., Kreimeyer, M., Lindemann, U. and Clarkson, P.J. (2009), "Reflecting communication: a key factor for successful collaboration between embodiment design and simulation", *Journal of Engineering Design*, Vol. 20 No. 3, pp. 265–287.

Maier, A.M. and Störrle, H. (2011), "What are the characteristics of engineering design processes?", *Proceedings of the 18th International Conference on Engineering Design (ICED 2011)*, Lyngby, pp. 188–198.

Maier, A.M., Wynn, D.C., Howard, T.J. and Andreasen, M.M. (2013), "Perceiving design as modelling: A cybernetic systems perspective", in Chakrabarti, A. and Blessing, L.T.M.

(Eds.),*An Ontology of theories and models of design*, pp. 133–148.

Malone, T.W. and Smith, S.A. (1988), “Modeling the Performance of Organizational Structures”, *Operations Research*, Vol. 36 No. 3, pp. 421–436.

March, J.G. and Sutton, R.I. (1997), “Crossroads—Organizational Performance as a Dependent Variable”, *Organization Science*, Vol. 8 No. 6, pp. 698–706.

Maurer, M.S. (2007), *Structural Awareness in Complex Product Design*, PhD Thesis, Technischen Universität München.

Meier, C., Yassine, A.A. and Browning, T.R. (2007), “Design Process Sequencing With Competent Genetic Algorithms”, *Journal of Mechanical Design*, Vol. 129 No. 6, p. 566.

Minai, A.A., Braha, D. and Bar-Yam, Y. (2006), “Complex Engineered Systems: A New Paradigm”, in Braha, D., Minai, A.A. and Bar-Yam, Y. (Eds.),*Complex Engineered Systems: Science Meets Technology*, Springer, Berlin, pp. 1–21.

Minneman, S.L. (1991), *The social construction of a technical reality: Empirical studies of group engineering design practice*, PhD Thesis, University of Stanford.

Moenaert, R.K., Caeldries, F., Lievens, A. and Wauters, E. (2000), “Communication Flows in International Product Innovation Teams”, *Journal of Product Innovation Management*, Vol. 17 No. 5, pp. 360–377.

Morelli, M.D., Eppinger, S.D. and Gulati, R.K. (1995), “Predicting technical communication in product development organizations”, *IEEE Transactions on Engineering Management*, Vol. 42 No. 3, pp. 215–222.

Morris, P.W.G. (1997), “Managing project interfaces - Key points for project success”, in Cleland, D.I. and King, W.R. (Eds.),*Project Management Handbook*, John Wiley & Sons, Inc., Hoboken, NJ, USA, pp. 16–55.

Morris, W.T. (1967), “On the Art of Modeling”, *Management Science*, Vol. 13 No. 12, pp. 707–717.

Moultrie, J. and Maier, A.M. (2014), “A simplified approach to design for assembly”, *Journal of Engineering Design*, Vol. 25 No. 1-3, pp. 44–63.

Newman, M.E.J. (2003), “The Structure and Function of Complex Networks”, *SIAM Review*, Vol. 45 No. 2, pp. 167–256.

Niloy, M., Andreas, G. and Animesh, D. (2009), *Dynamics On and Of Complex Networks: Applications to Biology, Computer Science, and the Social Sciences*, Springer, London.

Norman, D.O. and Kuras, M.L. (2006), “Engineering complex systems”, in Braha, D., Minai, A.A. and Bar-Yam, Y. (Eds.),*Complex Engineered Systems: Science Meets Technology*, Springer, Berlin, Vol. 21, pp. 206–245.

O’Donnell, F.J. and Duffy, A.H.B. (2002), “Modelling design development performance”, *International Journal of Operations & Production Management*, Vol. 22 No. 11, pp. 1198–1221.

Oliver, A.L. and Ebers, M. (1998), “Networking Network Studies: An Analysis of Conceptual Configurations in the Study of Interorganizational Relationships", *Organization Studies*, Vol. 19 No. 4, pp. 549–583.

Pahl, G., Beitz, W., Feldhusen, J. and Grote, K.-H. (2007), *Engineering Design: A Systematic Approach*, Springer, London.

Pappas, J.M. and Wooldridge, B. (2007), "Middle Managers' Divergent Strategic Activity: An Investigation of Multiple Measures of Network Centrality", *Journal of Management Studies*, Vol. 44 No. 3, pp. 323–341.

Parraguez, P., Eppinger, S.D. and Maier, A.M. (2014), "Evolution of information control and centralisation through stages of complex engineering design projects", in Dorian, M., Mario, S., Neven, P. and Nenad, B. (Eds.),*Proceedings of DESIGN 2014, the 13th International Design Conference*, Dubrovnik.

Pasqual, M.C. and de Weck, O. (2011), "Multilayer network model for analysis and management of change propagation", *Research in Engineering Design*, Vol. 23 No. 4, pp. 305–328.

Pektaş, Ş.T. and Pultar, M. (2006), "Modelling detailed information flows in building design with the parameter-based design structure matrix", *Design Studies*, Vol. 27 No. 1, pp. 99–122.

Phelps, C.C. (2010), "A Longitudinal Study of the Influence of Alliance Network Structure and Composition on Firm Exploratory Innovation", *Academy of Management Journal*, Vol. 53 No. 4, pp. 890–913.

PMI. (2014), *PMI's 2014 Pulse of the Profession: The High Cost of Low Performance*, PMI.

Poulin, R., Boily, M. and Masse, B.R. (2000), "Dynamical systems to define centrality in social networks."

Premkumar, G., Ramamurthy, K. and Saunders, C.S. (2005), "Information Processing View of Organizations: An Exploratory Examination of Fit in the Context of Interorganizational Relationships", *Journal of Management Information Systems*, Vol. 22 No. 1, pp. 257–294.

Provost, F. and Fawcett, T. (2013), *Data science for business: what you need to know about data mining and data-analytic thinking*, O'Reilly, Sebastopol.

Pullen, A.J.J., de Weerd-Nederhof, P.C., Groen, A.J. and Fisscher, O. a. M. (2012), "Open Innovation in Practice: Goal Complementarity and Closed NPD Networks to Explain Differences in Innovation Performance for SMEs in the Medical Devices Sector", *Journal of Product Innovation Management*, Vol. 29 No. 6, pp. 917–934.

Rahmani, K. and Thomson, V. (2011), "Managing subsystem interfaces of complex products", *International Journal of Product Lifecycle Management*, Vol. 5 No. 1, p. 73.

Raisch, S., Birkinshaw, J., Probst, G. and Tushman, M.L. (2009), "Organizational Ambidexterity: Balancing Exploitation and Exploration for Sustained Performance", *Organization Science*, Vol. 20 No. 4, pp. 685–695.

Reagans, R. and Zuckerman, E. (2001), "Networks, Diversity, and Productivity: The Social Capital of Corporate R&D Teams", *Organization science*, Vol. 12 No. 4, pp. 502–517.

Rechtin, E. (1990), *Systems Architecting: Creating & Building Complex Systems*, Prentice Hall, Michigan.

Rodan, S. and Galunic, C. (2004), "More than network structure: how knowledge heterogeneity influences managerial performance and innovativeness", *Strategic Management Journal*, Vol. 25 No. 6, pp. 541–562.

Rogers Commission. (1986), *Report on the Space Shuttle Challenger Accident*, Presidential Commission.

Roozenburg, N.F.M. and Cross, N.G. (1991), "Models of the design process: integrating across the disciplines", *Design Studies*, Vol. 12 No. 4, pp. 215–220.

Salvador, F. (2007), "Toward a Product System Modularity Construct: Literature Review and Reconceptualization", *IEEE Transactions on Engineering Management*, Vol. 54 No. 2, pp. 219–240.

Schön, D.A. (1984), *The Reflective Practitioner: How Professionals Think In Action*, Basic Books, New York.

Scott, J. (2000), *Social network analysis: a handbook*, SAGE Publications, London.

Senescu, R.R., Head, A.W., Steinert, M. and Fischer, M. (2012), "Generating a Network of Information Dependencies Automatically", *DSM 2012 Proceedings of the 14th International DSM Conference*, pp. 139–152.

Sharman, D.M. and Yassine, A.A. (2004), "Characterizing complex product architectures", *Systems Engineering*, Vol. 7 No. 1, pp. 35–60.

Sheard, S.A. and Mostashari, A. (2009), "Principles of Complex Systems for Systems Engineering", *Systems Engineering*, Vol. 12 No. 4, pp. 295–311.

Shears, P.A. (1971), "Engineering design as an information-processing activity", *Electronics and Power*, Vol. 17 No. 3, p. 100.

Shi, L., Gopsill, J.A., Snider, C., Jones, S., Newnes, L. and Culley, S.J. (2014), "Towards identifying pattern in engineering documents to aid project planning", in Dorian, M., Mario, S., Neven, P. and Nenad, B. (Eds.),*Proceedings of DESIGN 2014, the 13th International Design Conference*, Dubrovnik, pp. 1873–1882.

Sim, S.K. and Duffy, A.H.B. (2003), "Towards an ontology of generic engineering design activities", *Research in Engineering Design*, Vol. 14 No. 4, pp. 200–223.

Simon, H.A. (1946), "The Proverbs of Administration", *Public Administration Review*, Vol. 6 No. 1, p. 53.

Simon, H.A. (1962), "The architecture of complexity", *Proceedings of the American philosophical society*, Vol. 106 No. 6, pp. 467–482.

Simon, H.A. (1979), "Information processing models of cognition", *Annual review of psychology*, Vol. 30, pp. 363–396.

Simon, H.A. (1996), *The Sciences of the Artificial*, The MIT Press, Cambridge.

Smith, R.P. and Morrow, J.A. (1999), "Product development process modeling", *Design Studies*, Vol. 20 No. 3, pp. 237–261.

Sosa, M.E. (2000), *Analyzing the Effects of Product Architecture on Technical Communication in Product Development Organizations*, PhD Thesis, Massachusetts Institute of Technology.

Sosa, M.E. (2008), "A structured approach to predicting and managing technical interactions in software development", *Research in Engineering Design*, Vol. 19 No. 1, pp. 47–70.

Sosa, M.E. (2014), "Realizing the Need for Rework: From Task Interdependence to Social Networks", *Production and Operations Management*, Vol. 23 No. 8, pp. 1312–1331.

Sosa, M.E., Eppinger, S.D. and Rowles, C.M. (2003), "Identifying Modular and Integrative Systems and Their Impact on Design Team Interactions", *Journal of Mechanical Design*, Vol. 125 No. 2, p. 240.

Sosa, M.E., Eppinger, S.D. and Rowles, C.M. (2004), "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development", *Management Science*, Vol. 50 No. 12, pp. 1674–1689.

Sosa, M.E., Eppinger, S.D. and Rowles, C.M. (2007a), "A Network Approach to Define Modularity of Components in Complex Products", *Journal of Mechanical Design*, Vol. 129 No. 11, p. 1118.

Sosa, M.E., Eppinger, S.D. and Rowles, C.M. (2007b), "Are your engineers talking to one another when they should?", *Harvard Business Review*, No. November, pp. 133–142.

Sosa, M.E., Mihm, J. and Browning, T.R. (2011), "Degree Distribution and Quality in Complex Engineered Systems", *Journal of Mechanical Design*, Vol. 133 No. October, doi:10.1115/1.4004973.

Stephenson, K. and Zelen, M. (1989), "Rethinking centrality: Methods and examples", *Social Networks*, Vol. 11, pp. 1–37.

Sterman, J.D. (1994), "Learning in and about complex systems", *System Dynamics Review*, Vol. 10 No. 2-3, pp. 291–330.

Sterman, J.D. (2007), *Business dynamics: systems thinking and modeling for a complex world*, McGraw-Hill, Boston.

Stevenson, A. (Ed.). (2010), *Oxford Dictionary of English*, Oxford University Press, Oxford.

Steward, D. V. (1981), "The Design Structure System: A Method for Managing the Design of Complex Systems", *IEEE Transactions on Engineering Management*, Vol. 28 No. 3, pp. 71–74.

Storga, M., Stankovic, T., Cash, P. and McAloone, T.C. (2013), "Visually Augmented Analysis of Socio-Technical Networks in Engineering Systems Design Research", *Proceedings of TFTD13*, Luxembourg, p. 10.

Strogatz, S.H. (2001), "Exploring complex networks.", *Nature*, Vol. 410 No. 6825, pp. 268–76.

Sundararajan, A., Provost, F., Oestreicher-Singer, G. and Aral, S. (2013), "Information in digital, economic and social networks", *Information Systems Research*, Vol. 24 No. 4, pp. 883–905.

Talke, K., Salomo, S. and Kock, A. (2011), "Top Management Team Diversity and Strategic Innovation Orientation: The Relationship and Consequences for Innovativeness and Performance", *Journal of Product Innovation Management*, Vol. 28 No. 6, pp. 819–832.

Tan, P.-N. (2006), “Cluster Analysis: Basic Concepts and Algorithms”, *Introduction to Data Mining*, Pearson Addison Wesley, Boston, pp. 487–568.

The Guardian. (2014), “Investigation into 2010 BP oil spill finds failures, poor testing and ongoing risks”, *The Guardian*, available at: http://www.theguardian.com/environment/2014/jun/05/bp-deepwater-horizon-spill-report-failures-risks (accessed 5 June 2014).

Tortoriello, M., Reagans, R. and McEvily, B. (2011), “Bridging the Knowledge Gap: The Influence of Strong Ties, Network Cohesion, and Network Range on the Transfer of Knowledge Between Organizational Units”, *Organization Science*, Vol. 23 No. 4, pp. 1024–1039.

Tribelsky, E. and Sacks, R. (2010), “Measuring information flow in the detailed design of construction projects”, *Research in Engineering Design*, Vol. 21 No. 3, pp. 189–206.

Tsai, W. (2001), “Knowledge transfer in intraorganizational networks: effects of network position and absorptive capacity on business unit innovation and performance”, *Academy of Management Journal*, Vol. 44 No. 5, pp. 996–1004.

Turner, K.L. and Makhija, M. V. (2012), “The role of individuals in the information processing perspective”, *Strategic Management Journal*, Vol. 33, pp. 661–680.

Tushman, M.L., Katz, R. and Ralph, A. (1980), “External communication and project performance: an investigation into the role of gatekeepers”, *Management Science*, Vol. 26 No. 11, pp. 1071–1086.

Ulrich, K.T. (2011), *Design: Creation of Artifacts in Society*, University of Pennsylvania.

Ulrich, K.T. and Eppinger, S.D. (2012), *Product Design and Development*, McGraw-Hill Irwin, New York.

Van de Ven, A. (2011), *Engaged scholarship: a guide for organizational and social research*, Oxford University Press, Oxford.

Verma, J.P. (2012), *Data Analysis in Management with SPSS Software*, Springer, London.

Vignoli, M., Ferioli, G. and Steinert, M. (2013), “Building agile design teams”, in Lindemann, U., V, S., Kim, Y.S., Lee, S.W., Clarkson, J. and Cascini, G. (Eds.),*Proceedings of the 19th International Conference on Engineering Design (ICED13)*, Seoul, pp. 121–130.

Visser, W. (1992), “Designers’ activities at three levels: organization, strategies and problem-solving processes”, *Knowledge-Based Systems*, pp. 92–104.

Wang, D.J., Shi, X., McFarland, D. and Leskovec, J. (2012), “Measurement error in network data: A re-classification”, *Social Networks*, Elsevier B.V., Vol. 34, pp. 396–409.

Wasserman, S. and Faust, K. (1989), “Canonical Analysis of the Composition and Structure of Social Networks”, *Sociological methodology*, Vol. 19, pp. 1–42.

Wasserman, S. and Faust, K. (1994), *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge.

De Weck, O. (2007), “On the role of DSM in designing systems and products for changeability", *Proceedings of 9th International Design Structure Matrix Conference*, Munich, Vol. 1, pp. 311–323.

De Weck, O., Roos, D. and Magee, C.L. (2011), *Engineering Systems: Meeting Human Needs in a Complex Technological World*, The MIT Press, Cambridge.

Weisstein, E. (2003), *CRC concise encyclopedia of mathematics*, Chapman & Hall/CRC, Boca Raton.

Wyatt, D.F., Wynn, D.C. and Clarkson, P.J. (2013), "A Scheme for Numerical Representation of Graph Structures in Engineering Design", *Journal of Mechanical Design*, Vol. 136 No. 1, p. 011010.

Wyatt, D.F., Wynn, D.C., Jarrett, J.P. and Clarkson, P.J. (2012), "Supporting product architecture design using computational design synthesis with network structure constraints", *Research in Engineering Design*, Vol. 23 No. 1, pp. 17–52.

Wynn, D.C. (2007), *Model-Based Approaches to Support Process Improvement in Complex Product Development*, PhD Thesis, University of Cambridge.

Wynn, D.C., Caldwell, N.H. and Clarkson, P.J. (2014), "Predicting Change Propagation in Complex Design Workflows", *Journal of Mechanical Design*, Vol. 136 No. August, doi:10.1115/1.4027495.

Wynn, D.C., Eckert, C.M. and Clarkson, P.J. (2006), "Applied Signposting: a modeling framework to support design process improvement", *ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering*, Philadelphia.

Yassine, A.A., Chelst, K.R. and Falkenburg, D.R. (1999), "A Decision Analytic Framework for Evaluating Concurrent Engineering", *IEEE Transactions on Engineering Management*, Vol. 46 No. 2, pp. 144–157.

Yassine, A.A., Falkenburg, D.R. and Chelst, K. (1999), "Engineering design management: An information structure approach", *International Journal of Production Research*, Vol. 37 No. 13, pp. 2957–2975.

Yassine, A.A., Joglekar, N., Braha, D., Eppinger, S.D. and Whitney, D.E. (2003), "Information hiding in product development: the design churn effect", *Research in Engineering Design*, Vol. 14 No. 3, pp. 145–161.

Yassine, A.A., Sreenivas, R.S. and Zhu, J. (2008), "Managing the exchange of information in product development", *European Journal of Operational Research*, Vol. 184 No. 1, pp. 311–326.

Yassine, A.A., Whitney, D.E., Daleiden, S. and Lavine, J. (2003), "Connectivity maps: Modeling and analysing relationships in product development processes", *Journal of Engineering Design*, Vol. 14 No. 3, pp. 377–394.

Yassine, A.A. and Wissmann, L.A. (2007), "The Implications of Product Architecture on the Firm", *Systems Engineering*, Vol. 10 No. 2, pp. 118–137.

Yin, R. (2009), *Case study research: design and methods*, SAGE, London.

APPENDICES

A. Glossary of terms

B. Electronic questionnaire

C. List of people interviewed

D. Structured questionnaire

E. Examples of annotated process pictures (exploratory case study)

F. Examples of gathered material

G. Net-Sights 1.0 Overview

H. NPr Framework workflow

I. Equations for selected network architecture metrics

A. Glossary of terms

- **Activity category:** The highest (most aggregated) level in the hierarchical process breakdown used in this thesis.
- **Activity group:** The level in the hierarchical process breakdown below 'activity category', and based on cohesive work-packages associated with each subsystem that is being designed.
- **Actual design process:** The design process that happens and that is observable through expressed behaviour.
- **Architecture (network or system architecture):** The combination of compositional and structural network characteristics of a system or network.
- **Complex(ity):** A characteristic associated with something that 1) contains multiple elements, 2) possesses a number of connections between the elements, 3) exhibits dynamic interactions between the elements, and 4) exhibits behaviour that cannot be explained by the simple sum of its elements.
- **Component:** 'A component of a system is a subset of the physical realisation (and the physical architecture) of the system to which a subset of the system's functions have been (will be) allocated' (Buede, 2009, p. 61).
- **Design activity:** A design activity is a construct that refers to the actual realisation of a particular design task. It involves actions executed individually or in a team to transform a set of information inputs into a set of information outputs.
- **Design object:** 'The object treated in the design process, the object process and/or system being designed' (Hubka, 1996, p. 83).
- **Design process model:** 'A design-process model is an attempt to describe a real design process in an abstract way. Models must make choices about how and to what extent to abstract from reality. Such decisions should align with the purposes or intended uses of the model (of which there may be many). Hence, different modellers may produce very different descriptions of the same design process' (Clarkson and Eckert, 2005, p. 62).
- **Design process:** 'The network of activities performed with the goal of producing a design' (Clarkson and Eckert, 2005, p. 61). 'A design process is a real, actual way in which design

work is done and designs are produced. A design-process model is an attempt to describe a real design process in an abstract way' (Clarkson and Eckert, 2005, p. 62).

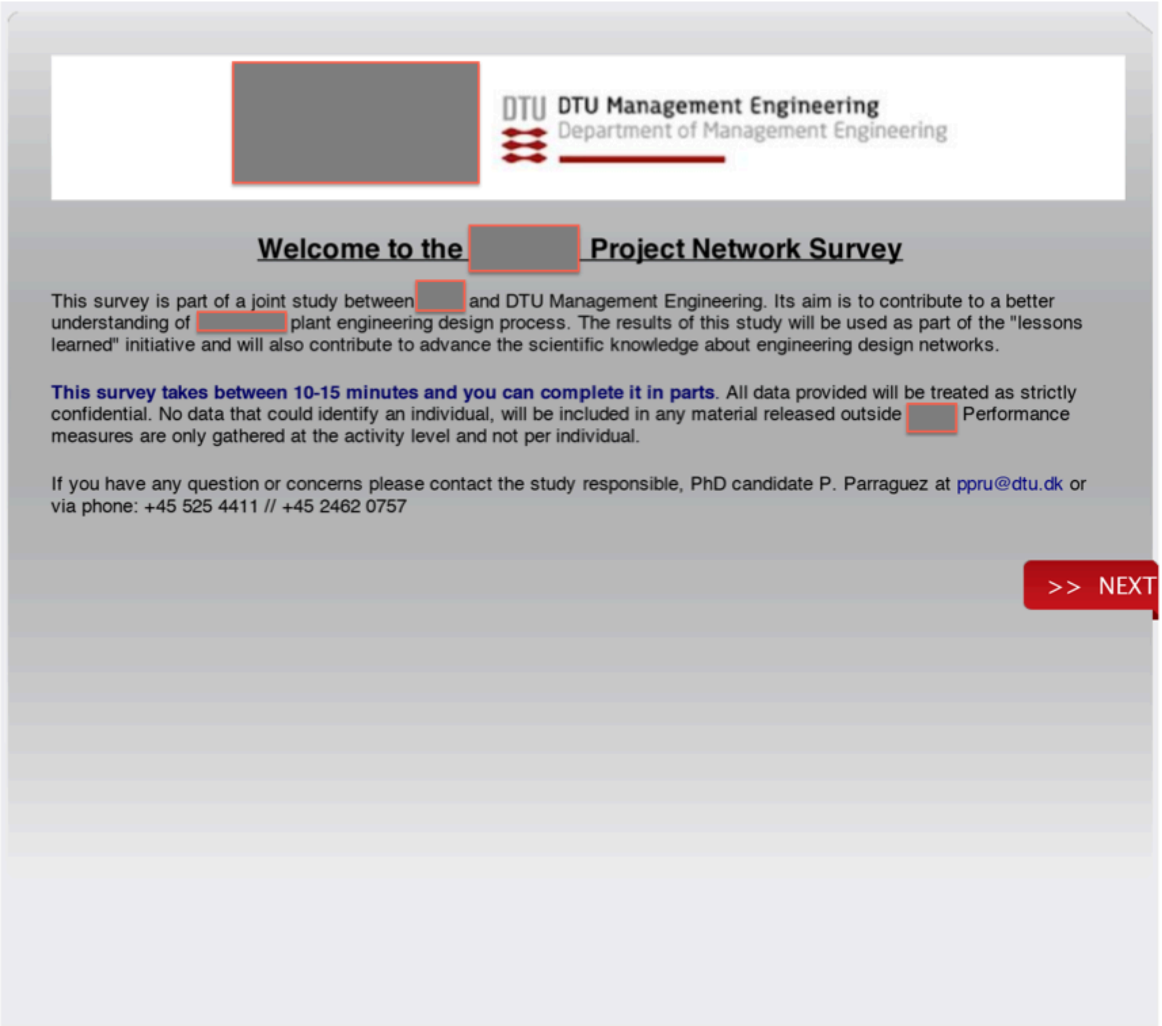
- **Design task:** A design task is the work that is required or specified in order to achieve a particular design objective. The objective of a design task can relate to the definition or evaluation of a parameter in the design object, or to the management of the design process.
- **Designer:** 'Designer', 'design engineer', and 'engineering designer' are used as equivalent to denote anybody who performs a design activity in the scope of the engineering design project under analysis.
- **Domain**: Specific view of a complex system, comprising one type of entity (Kreimeyer and Lindemann, 2011, p. 37).
- **Edge level:** Level at which the relationship between two elements is located, such as the edge that represents the information dependency between two tasks.
- **Engineering Design:** Research field that studies 'the process of converting an idea or market need into the detailed information from which a product or technical system can be produced' (Hales and Gooch, 2004, p. 2).
- **Engineering system:** 'A class of systems characterized by a high degree of technical complexity, social intricacy, and elaborate processes, aimed at fulfilling important functions in society' (de Weck et al., 2011, p. 31).
- **Engineering Systems:** Research field that studies engineering systems.
- **Information:** In engineering design, **information** consists of a combination of design inputs and outputs that can take the form of written documents, conversations, visual representations, gestures, and so on (Maier, Kreimeyer, et al., 2009). In the context of a design activity, this information has the purpose of defining the design object, evaluating design options, and/or coordinating the design process (Sim and Duffy, 2003).
- **Information exchange:** A communication event in which an information package containing a set of information items (Tribelsky and Sacks, 2010) is transmitted between parties of the design process at a particular point in time.
- **Information flow:** A combination of information exchanges; more precisely, a set of information packages exchanged between designers from one design activity to another over a period of time (Tribelsky and Sacks, 2010).

- **Information item:** A single piece of information containing data about a design parameter, such as dimensions, weight, amount (Tribelsky and Sacks, 2010).
- **Information package:** set of related information items that can take the form of a drawing, a worksheet, a document, a presentation, and so on (Tribelsky and Sacks, 2010).
- **Information transformation:** The activity by which meaning is assigned to information inputs, subsequently transforming them into knowledge (Bruce and Cooper, 2000). In the context of design activities, **information transformation** allows abstract statements of requirements to become detailed specifications of a product, usually in the form of graphic and textual representations (Chira, 2005; Culley, 2014; Hubka, 1996; Shears, 1971).
- **Interface:** 'An interface is a connection resource for hooking to another system's interface (an external interface) or for hooking one system's component to another (an internal interface). Interfaces have inputs, produce outputs, and perform functions. An interface can be as simple as a wire or conveyor belt or as sophisticated as a global communication system (which is a system in its own right)' (Buede, 2009, p. 61)**.**
- **Model:** A model is 'an approximation, representation, or idealization of selected aspects of the structure, behavior, operation, or other characteristics of a real-world process, concept, or system' (IEEE Standards Board, 1989, p. 12), that is, a model is an abstraction.
- **Model view:** A model view is a representation of a system from the perspective of specific concerns or issues (IEEE Standards Board, 2000, p. 3).
- **Network:** A set of interconnected elements.
- **Network composition:** The combination of elements of a system. Through a system's composition we obtain information about the nature of the different elements being connected (Wasserman and Faust, 1994). Alongside network structure is what constitutes the network architecture.
- **Network structure**: The arrangement of and relationships between elements of a system. Through a system's structure we obtain how things connect to each other (Wasserman and Faust, 1994). Alongside network composition is what constitutes the network architecture.
- **Networked:** Term used to refer to what has an interconnected nature that can be described as a network.

- **Node level:** The level at which the individual element that is part of a network is located, such as the node that represents a task in a design process model.
- **Organisation architecture:** ‘The structure of an organization embodied in its people, their relationships to each other and to the organization’s environment, and the principles guiding its design and evolution. Organization architectures generally group people into teams, departments, or other types of organizational units. The terms organization architecture and organization structure are often used interchangeably, although the latter term is also used in the more limited sense of lines of authority (reporting relationships)’ (Eppinger and Browning, 2012, p. 80).
- **Organisation domain:** Refers to the organisation or group of organisations and other stakeholders involved in the design process of an engineering system and has an architecture defined by people and their interactions.
- **Planned design process:** The design process envisioned or intended.
- **Process:** ‘A series of actions or steps taken in order to achieve a particular end’ (Stevenson, 2010).
- **Process architecture:** The structure of activities and their relationships (Browning, 2009).
- **Process domain**: The series of activities taken or tasks needed to design the engineering system.
- **Product:** In this thesis, the product is the engineering system being designed. A ‘product’ as an engineering system is typically composed of several subsystems, each of which may comprise one or more components (e.g. Yassine & Wissmann 2007; Salvador 2007).
- **Product architecture:** ‘Defines the functional elements within an artefact [e.g. engineering system], maps these functional elements to physical elements, and defines the interfaces among the interacting physical elements’ (Yassine and Wissmann, 2007).
- **Product domain:** The engineering system being designed, including its architecture, which is defined by its technical components and how the components are interconnected.
- **System:** ‘A combination of interacting elements organized to achieve one or more stated purposes’ (IEEE Standards Board, 2000), and less generically, ‘a set of components (subsystems, segments) acting together to achieve a set of common objectives via the accomplishment of a set of tasks’ (Buede, 2009, p. 50).

- **System architecture:** ‘The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment’ (IEEE Standards Board, 2000). ‘The structure of a system embodied in its elements, their relationships to each other (and to the system’s environment), and the principles guiding its design and evolution that gives rise to its functions and behaviors’ (Eppinger and Browning, 2012, p. 7).
- **System function:** A ‘set of functions that must be performed to achieve a specific objective’ (Buede, 2009, p. 50).
- **Whole process level:** The level that describes the whole network architecture, including nodes and edges.

B. Electronic questionnaire



Please select your **departmental affiliation(s)**

*Include **all** the departments and resource groups to which you have been allocated **since the beginning of the** [redacted] **project.***

- ☐ 100 - CEO
- ☐ ▹ 110 - QHSE
- ☐ ▹ 120 - QA/QC
- ☐ 200 - CFO
- ☐ ▹ 210 - Economy and Finance
- ☐ ▹ 220 - HR & Gen. Services
- ☐ ▹ 230 - Information Technology
- ☐ 320 - Sales & Marketing
- ☐ ▹ 321 - Proposal
- ☐ 340 - Operations
- ☐ ▹ 341 - Procurement & Logistics
- ☐ ▹ 342 - Project Management
- ☐ ▹ 343 - Planning
- ☐ 360 - Engineering
- ☐ ▹ 361 - Process and Structural Mechanics
- ☐ ▹ 362 - Mechanical
- ☐ ▹ 363 - E and I&C
- ☐ ▹ 364 - Auxiliaries and Retrofit
- ☐ 380 - Service
- ☐ ▹ 381 - Service Sales
- ☐ ▹ 382 - Service Project
- ☐ ▹ 383 - Site Services

<< BACK   >> NEXT

Please select all the **activities where you participate in the context of the [redacted] project**

Please select all the areas/activities/sub-systems in which you participate directly (providing inputs or receiving requests for example), not just your formal functional affiliation.

- [ ] Overall Project Management
  (Includes overall design approval and other general activities at the project level)
- [ ] Procurement (including RFQs)
- [x] Load Plan and Layout
- [ ] Boiler and Equipment Design
- [ ] PFD + P&ID
- [x] External Piping
- [ ] COMOS Data
- [ ] Steel Related Activities
- [x] Pressure Parts Design
- [ ] Air and Flue Gas
- [ ] Combustion System
- [ ] Electrical, Control and Instrumentation

<< BACK   >> NEXT

What is the **responsibility you perceive you have** in the following activities?

Please indicate the level of responsability that you think you have (not necessarily what is formally described)

	Low responsability *You contribute to the activity but you don't think you are the activity responsable*	**Medium responsability** *You contribute to the activity and you think you are somewhat responsable (but not the main responsable)*	**High Responsability** *You contribute to the activity and you think you are the main activity responsable.*
Load Plan and Layout	○	○	○
External Piping	○	○	○
Pressure Parts Design	○	○	○

The following questions ask for your **intuitive assessment of the activities in which you have participated.** You are not asked to provide an accurate and objective value, only a subjective and quick assessment from your personal point of view.

Please assess the **resource efficiency** (in terms of time and budget) of the activities where you participate

If you are unsure please provide your best estimate or simply skip the activity where an estimate is not possible

	Significantly below expectations	Below expectations	On expectations	Above expectations	Significantly above expectations.
	1	2	3	4	5
Load Plan and Layout					
External Piping					
Pressure Parts Design					

Please assess the **quality of the results** (in terms of the desired technical specifications) of the activities where you participate

If you are unsure please provide your best estimate or simply skip the activity where an estimate is not possible

	Significantly below expectations	Below expectations	On expectations	Above expectations	Significantly above expectations.
	1	2	3	4	5
Load Plan and Layout					
External Piping					
Pressure Parts Design					

Please assess the **innovativeness of the activity results**

- Activity results are innovative when they produce a positive output believed to be novel/original inside of the organisation.

- If you are unsure please provide your best estimate or simply skip the activity where an estimate is not possible

	Significantly below expectations 1	Below expectations 2	On expectations 3	Above expectations 4	Significantly above expectations. 5
Load Plan and Layout					
External Piping					
Pressure Parts Design					

Please assess the **overall performance** of the activities

If you are unsure please provide your best estimate

	Significantly below expectations 1	Below expectations 2	On expectations 3	Above expectations 4	Significantly above expectations. 5
Load Plan and Layout					
External Piping					
Pressure Parts Design					

<< BACK

>> NEXT

Please select **who you have interacted with inside** [redacted] in the context of the [redacted] plant

- An interaction here is defined as work related direct ***information exchanges necessary to do one or more project activities*** *(This includes emails, work conversations and other forms of work communications)*

- Initials added in blue after the "+" symbol represent people who were previously performing the same function but who are not in that position anymore and/or are now providing support to the main person currently performing this function.

[redacted]

Please quantify the **impact, direction and frecuency of your interactions**

*- Consider the **combination of activities in which you interact with that person***

*- An interaction here is defined as work related direct **information exchanges necessary to do one or more project activities** (This includes emails, work conversations and other forms of work communication)*

	Impact of the Interaction on the performed activities			**Direction** of the interaction			**Frecuency** of your interactions		
	Interaction has a low impact (+)	Interaction has medium impact (++)	Interaction has high impact (+++)	Usually I initiate the interaction	Interactions tend to be mutually initiated	Usually the other party initiates the interaction	Daily	Weekly	Monthly
[redacted]	○	○	○	○	○	○	○	○	○
	○	○	○	○	○	○	○	○	○
	○	○	○	○	○	○	○	○	○
	Interaction has a low impact (+)	Interaction has medium impact (++)	Interaction has high impact (+++)	Usually I initiate the interaction	Interactions tend to be mutually initiated	Usually the other party initiates the interaction	Daily	Weekly	Monthly
	○	○	○	○	○	○	○	○	○
	○	○	○	○	○	○	○	○	○

<< BACK   >> NEXT

Please select **who you have interacted with outside** [redacted] in the context of the [redacted] project

*- An interaction here is defined as work related direct **information exchanges necessary to do one or more project activities** (This includes emails, work conversations and other forms of work communications)*

-The list contains key organisations and individuals. If you have a relevant interaction outside [redacted] that is not covered here, please write the organisation's name at the end of this list (using one of the "other" fields).

- ☐ [redacted]
- ☐
- ☐
- ☐
- ☐
- ☐
- ☑
- ☐
- ☐
- ☐

Please quantify the **impact, direction and frecuency of your interactions**

*- Consider the **combination of activities in which you interact with that person***

*- An interaction here is defined as work related direct **information exchanges necessary to do one or more project activities** (This includes emails, work conversations and other forms of work communication)*

	Impact of the Interaction on the performed activities			**Direction** of the interaction			**Frequency** of your interactions		
	Interaction has a low impact (+)	Interaction has a medium impact (+)	Interaction has a high impact (+++)	Usually I initiate the interaction	Interactions tend to be mutually initiated	Usually the other party initiates the interaction	Daily	Weekly	Monthly
	○	○	○	○	○	○	○	○	○
	○	○	○	○	○	○	○	○	○
	○	○	○	○	○	○	○	○	○
	Interaction has a low impact (+)	Interaction has a medium impact (+)	Interaction has a high impact (+++)	Usually I initiate the interaction	Interactions tend to be mutually initiated	Usually the other party initiates the interaction	Daily	Weekly	Monthly
	○	○	○	○	○	○	○	○	○
	○	○	○	○	○	○	○	○	○

<< BACK

>> NEXT

Project level performance evaluation

Based on your personal assessment of the project as a whole

	Significantly below expectations	Below expectations	On expectations	Above expectations	Significantly above expectations
	1	2	3	4	5
Resource **efficiency** (project on time and on budget)					
Please assess the **quality of the results** (in terms of the desired technical specifications) for the project as a whole					
Perceived **innovativeness** of the project results					
Overall performance assessment					

Please indicate your **knowledge of the project as whole**

This refers to your level of overview of the different activities, the activities dependencies and the different interactions in the context of [redacted] plant beyond your assigned activities and functional area

	Very low	Low	High	Very high
	1 2	3	4	5
Completeness of your project level overview				

Are you **satisfied with your current level of knowledge of the project as a whole?**

This refers to your satisfaction related to your answer in the previous question

	Very Dissatisfied	Dissatisfied	Somewhat Dissatisfied	Neutral	Somewhat Satisfied	Satisfied	Very Satisfied
	1	2		3		4	5
Satisfaction about your project level overview							

<< BACK

>> NEXT

C. List of people interviewed

Exploratory case study, list of structured interviews:

- CTO
- Project Manager Membranes
- Production Manager
- Project and Quality Manager
- Business Manager 1
- Business Manager 2
- Engineering Intern
- Sales & Marketing Coordinator
- R&D Engineer 1
- R&D Engineer 2

Descriptive case study, open and semi-structured interviews:

- Vice President of Operations
- Vice President of Engineering
- Technical Project Manager
- QA/QC Manager
- QA/QC Engineer
- Site Manager
- Procurement Manager

D. Guided questionnaire

Questionnaire used in the exploratory case study

Networked Engineering Design Activities – Structured Questionnaire

1) Personal and project level information

Full Name	
Age	
Position	
Department & Group	
Previous Position	
Years in the Company	
Previous Work	
LinkedIn?	
Overall Project Assessment	
• On time	
• On Budget	
• On Spec	
• Quality	
• Innovativeness	
Network Awareness	

2) Design Activities

Activity Code	Defined role in the activity	Relative time requirement	Performance *Significantly below (1), below (2), on expectation (3), above (4) or significantly above expectations (5).*					Overall Perfor-mance
Code from table	Leader (3), responsible (2), supporter (1)	In relation to the time spent in all other activities: *Above, around or below average*	Efficiency			Effectiveness		Subjective. On same scale as before
			On time	On Budget	On Spec	Quality	Innovativeness	

3) Design Activities Interactions

Person Code	Activities in which interaction exists	Proportion of time spent in those activities
Code from table	Activity code(s)	Out of total interactions both social and job related

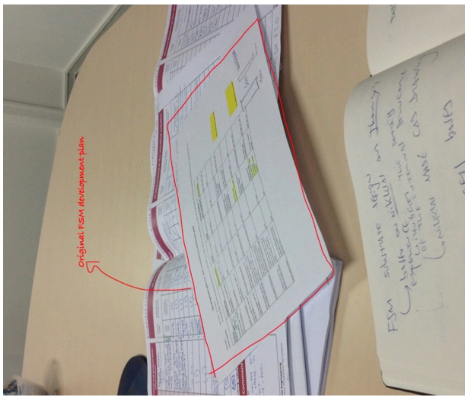
4) Comments

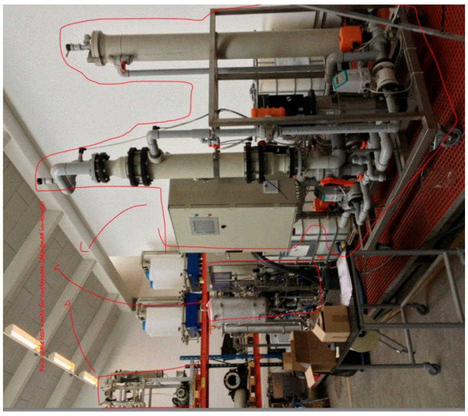
DTU Management Engineering
Institut for Systemer, Produktion og Ledelse

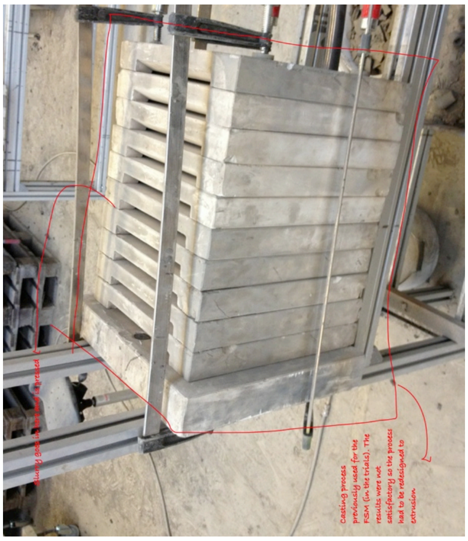
Pedro Parraguez Ruiz – 22 Nov 2012

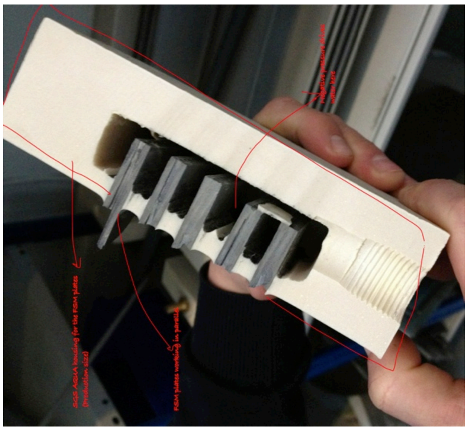
E. Annotated process pictures

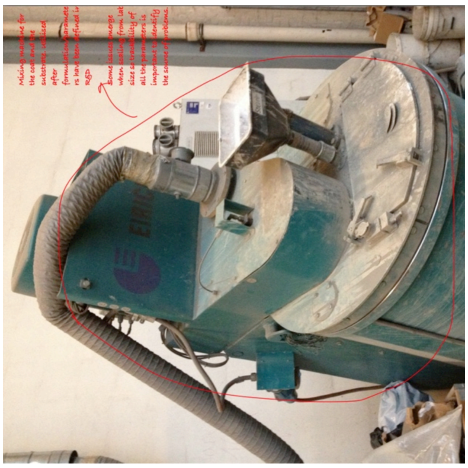
Examples from the exploratory case study, capturing the design process of the flat-sheet membrane

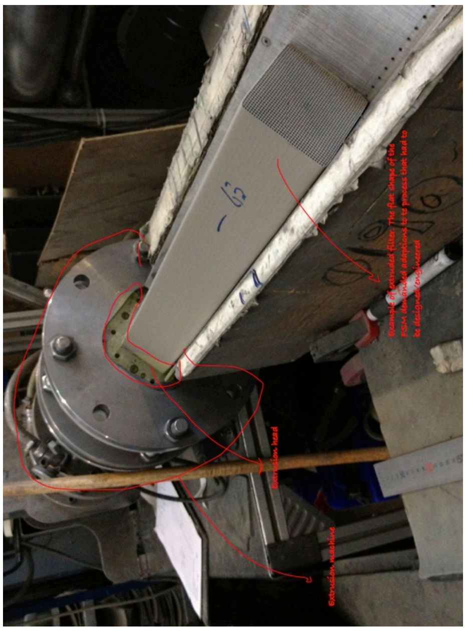




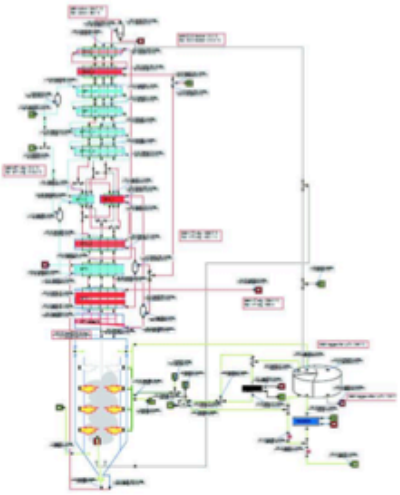


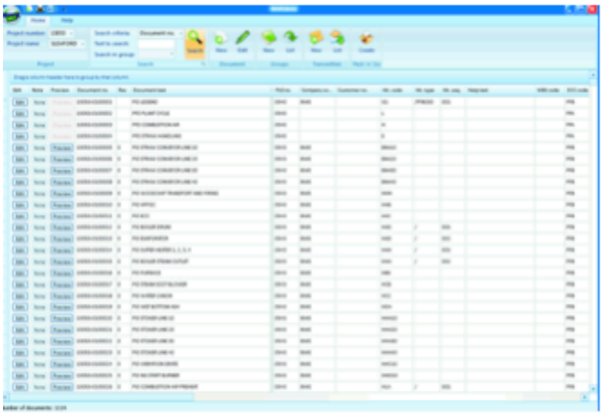






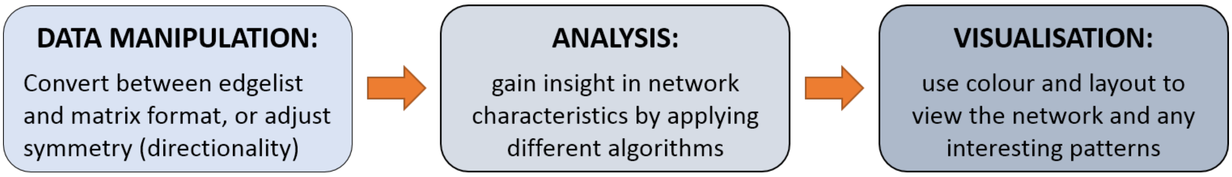
F. Examples of gathered material



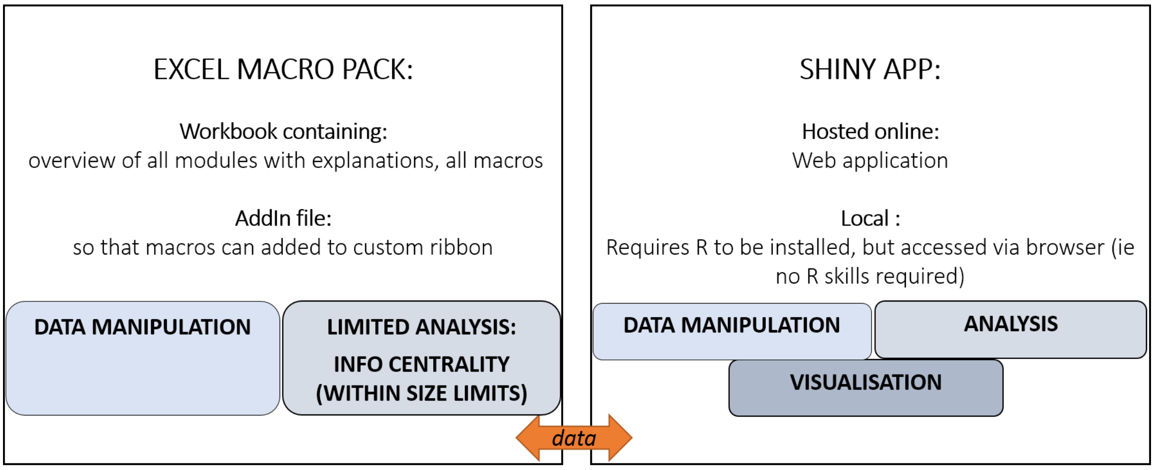


G. Net-Sights 1.0 overview

The **Net-Sights 1.0** is an application created to streamline some parts of the network analysis and visualisation work required by the NPr Framework. It has a series of modules that can be used to perform common tasks. The basic stages include data manipulation, analysis, and visualisation:



There are two solutions available: an Excel workbook containing a set of useful macros, and a web application deployed through a combination of RStudio and Shiny.

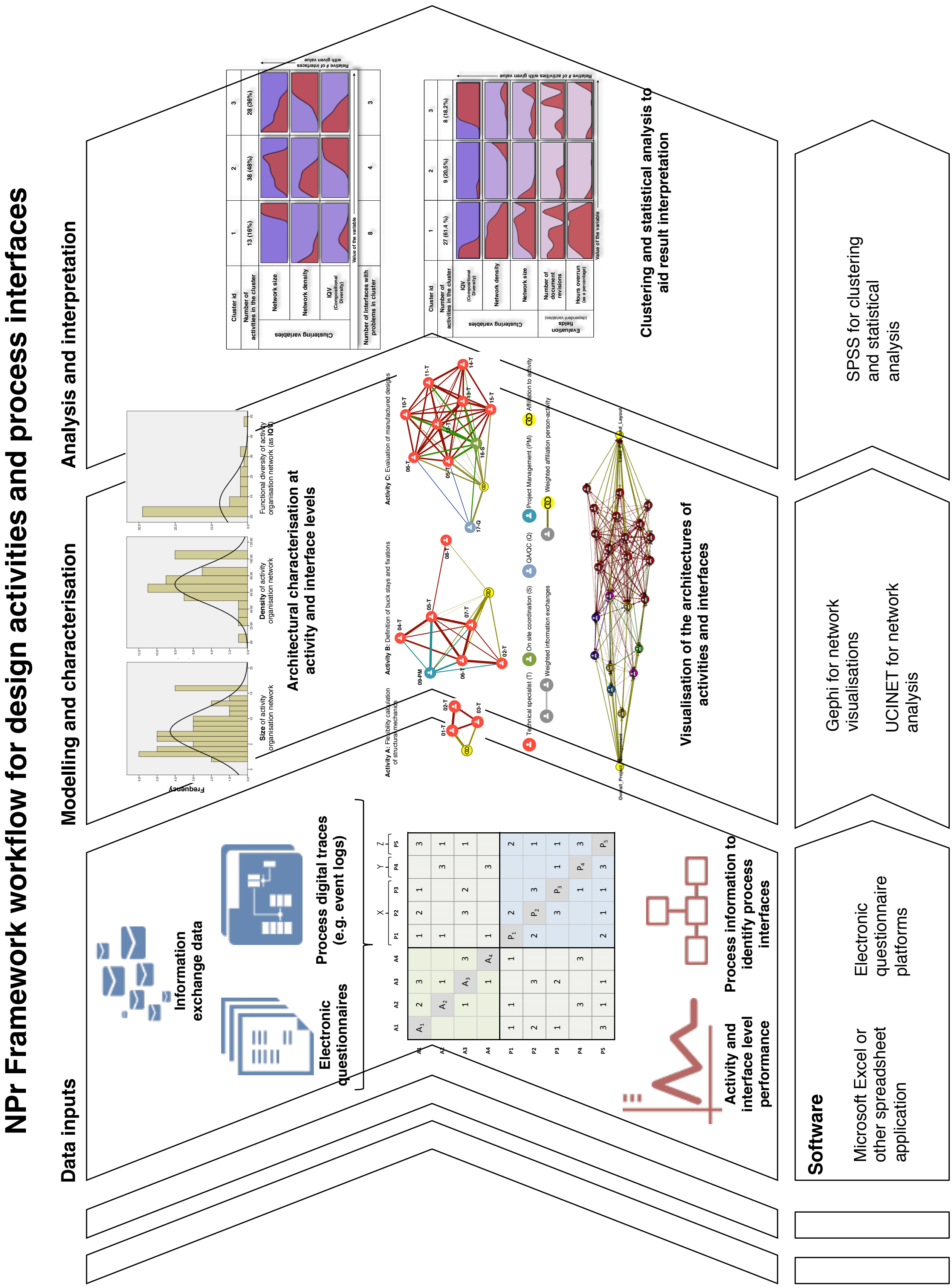


The **Excel macro pack** is a set of macros that can be used in Excel to facilitate common data manipulation tasks. This solution is limited by restrictions in Excel that affect the extent to which data can be analysed, but offers the convenience of being based in Excel, where data is commonly stored and edited. It is primarily intended for use among students/researchers who already work with data inside Excel and seek to streamline their processes.

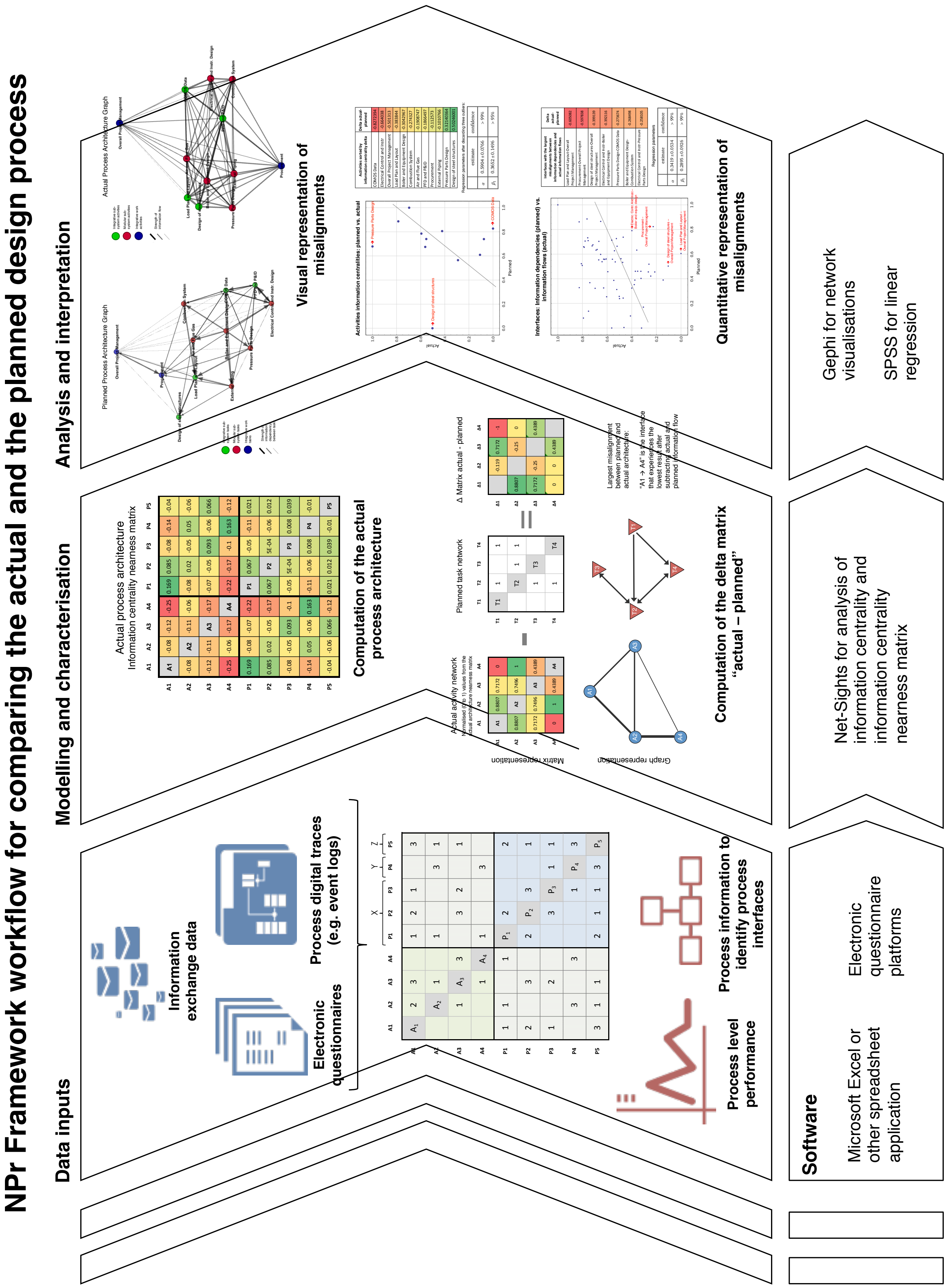
The **Shiny application** is a browser-based application created using the R software package. This solution can perform more extensive analysis tasks as well as basic network visualisation. It can be used by almost anyone (including users who do not regularly use Excel), and therefore, is more suitable for practioners. The Shiny application is available online at http://bit.ly/ESG-NetSights.

A more detailed description of each component is provided in the guide available at http://bit.ly/ESG-NetSights, including instructions for installation and use.

H. NPr Framework workflow – Activities and process interfaces

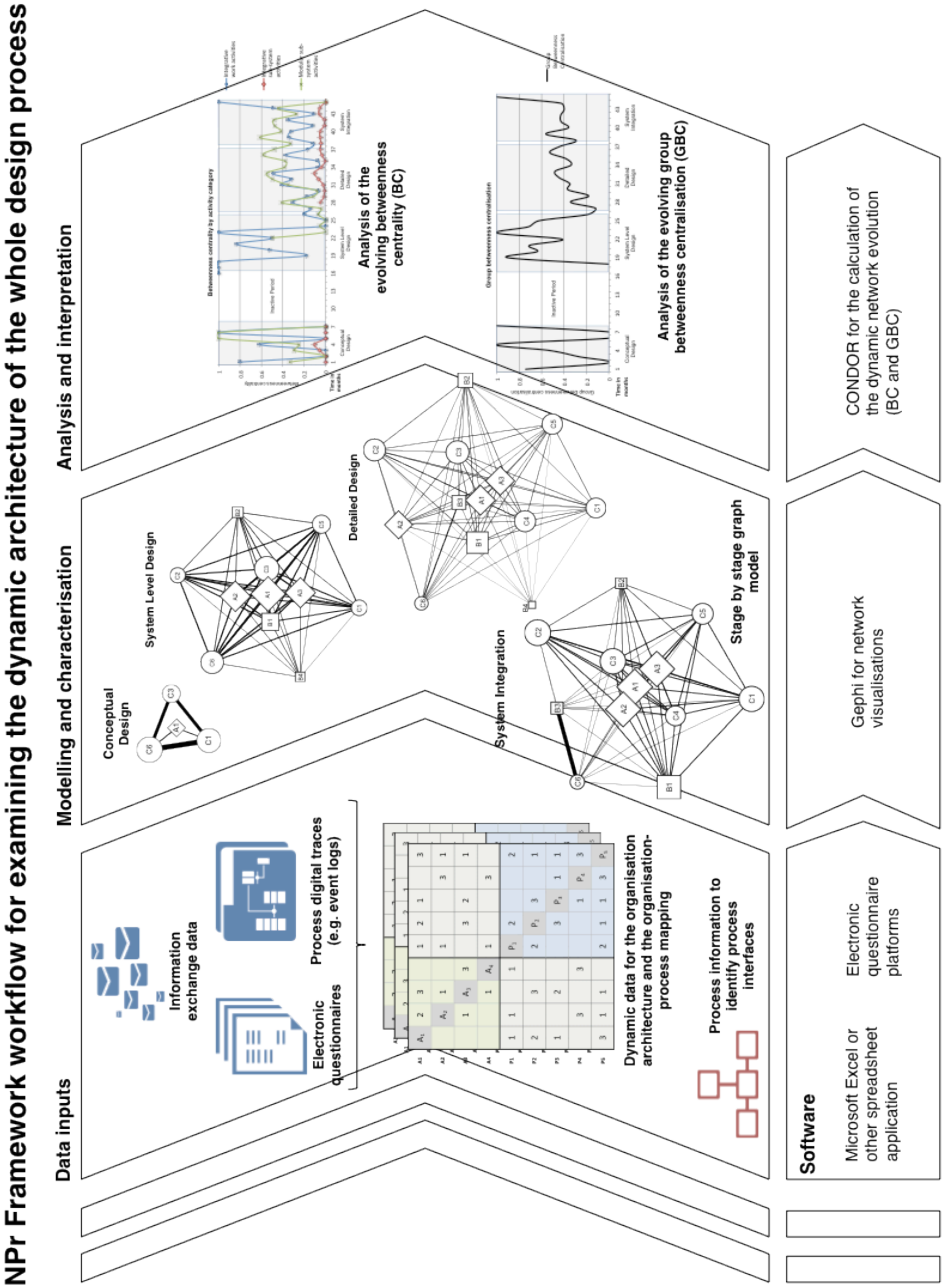


NPr Framework workflow – Whole process comparison of actual and planned architecture



NPr Framework workflow – Whole process

Dynamic architecture



I. Equations for selected network architecture metrics

Betweenness centrality

'Betweenness centrality is defined as the share of times that a node i needs a node k (whose centrality is being measured) in order to reach a node j via the shortest path. Specifically, if g_{ij} is the number of geodesic paths from i to j, and g_{ikj} is the number of these geodesics that pass through node k, then the betweenness centrality of node k is given by':

$$\sum_i \sum_j \frac{g_{ikj}}{g_{ij}}, \quad i \neq j \neq k$$

(Borgatti, 2005, p. 60)

Group betweenness centrality

'Group betweenness centrality measures the proportion of geodesics connecting pairs of nongroup members that pass through the group. Let C be a subset of nodes of a graph with node set V, let $g_{u,v}$ be the number of geodesics connecting u to v, and let $g_{u,v}(C)$ be the number of these geodesics that pass through C. Then the group betweenness centrality of C is given by:

$$\text{Group betweeness centrality} = \sum_{u<v} \frac{g_{u,v}(C)}{g_{u,v}} \quad u, v \notin C.$$

This value can then be normalized by dividing by 1/2 $(|V| - |C|)(|V| - |C| - 1)$, which is the maximum possible.

$$\text{Normalized group betweeness centrality} = \frac{2 \sum_{u<v} \frac{g_{u,v}(C)}{g_{u,v}}}{(|V| - |C|)\,(|V| - |C| - 1)}$$

where u,v $\in / C$ '

(Carrington et al., 2005, p. 62)

Stephenson and Zelen information centrality

'Let **A** be an adjacency matrix describing the connected network, **D** a diagonal matrix of the degree of each point and **J** a matrix with all its elements equal to one. The index of centrality of Stephenson and Zelen is calculated by inverting the matrix **B** defined by:

$\mathbf{B} = \mathbf{D} - \mathbf{A} + \mathbf{J}$,

in order to obtain the matrix:

$$\mathbf{C} = (c_{ij}) = \mathbf{B}^{-1}$$

from which the information matrix is given explicitly by:

$$\mathbf{I}_{ij} = \left(c_{ii} + c_{jj} - 2c_{ij}\right)^{-1}$$

The values $\mathbf{I}_{ij}$ summarize the information contained in all possible paths between points i and j.

To define the centrality index associated with the point i, Stephenson and Zelen use the harmonic average ':

$$C_{\mathrm{Inf}}(i) = \left[\frac{1}{N}\sum_j \frac{1}{\mathbf{I}_{ij}}\right]^{-1}$$

(Poulin et al., 2000, p. 196; Stephenson and Zelen, 1989)

Index of Qualitative Variation (IQV)

IQV index is calculated as follows:

$$IQV = \frac{K(100^2 - \sum Pct^2)}{100^2(K-1)}$$

where

K = the number of categories (for example, the total number of departments), and

$\sum Pct^2$ = the sum of all square percentages in the distribution (as an integer number).

This page intentionally left blank