SOCIO-TECHNICAL ANALYSIS OF SYSTEM-OF-SYSTEMS USING RESPONSIBILITY MODELLING

David Greenwood

A Thesis Submitted for the Degree of PhD
at the
University of St. Andrews

2012

Socio-Technical Analysis of System-of-Systems Using Responsibility Modelling

David Greenwood

This thesis is submitted in partial fulfilment for the degree of PhD
at the
University of St Andrews

7th June 2012

Socio-Technical Analysis of System-of-Systems Using Responsibility Modelling

David Greenwood

School of Computer Science,

University of St Andrews

A thesis submitted to the University of St. Andrews for the degree of PhD.

Keywords: system-of-systems; socio-technical systems engineering; responsibility modelling; threat identification; troubleshooting.

Declaration: No portion of the work in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning. Portions of this thesis have appeared previously in the following published articles.

Publications:

1. **Greenwood**, Sommerville (2011) "Expectations and Reality: Why an enterprise software system didn't work as planned", 20th International Conference on Information Systems Development, Edinburgh
2. **Greenwood**, Sommerville (2011) "Using Complex Network Analysis and Visualisation to Analyse Problematic Enterprise Scale Information Systems?", 55th Meeting of the International Society for the Systems Sciences, Hull
3. **Greenwood**, Sommerville (2011) "Responsibility Modeling for Identifying Sociotechnical Threats to the Dependability of Coalitions of Systems", 6th IEEE International Conference on Systems-of-Systems Engineering, Albuquerque
4. Khajeh-Hosseini, **Greenwood**, Sommerville (2010), "Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS," IEEE 3rd International Conference on Cloud Computing, Miami

Candidate's Declarations

I, David Greenwood hereby certify that this thesis, which is approximately 70,000 words in length, has been written by me, that it is the record of work carried out by me and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in September, 2008 and as a candidate for the degree of PhD in September, 2009; the higher study for which this is a record was carried out in the University of St Andrews between 2008 and 2012.

Date 07/06/2012 signature of candidate ………

Supervisor's Declaration

I, Professor Ian Sommerville, hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date 07/06/2012 signature of supervisor ………

Permission for Electronic Publication

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the electronic publication of this thesis:

(i) Access to printed copy and electronic publication of thesis through the University of St Andrews.

Date 07/06/2012

signature of candidate ……

signature of supervisor ………

Acknowledgements

This work would not have been possible without the help, generosity and encouragement of many people.

I would like to thank my supervisor, Ian Sommerville, for his patience, inspiration and guidance throughout my studies at the University of St Andrews. I am grateful to the many members of LSCITS project, and particularly Dave Cliff (LSCITS project director), for providing memorable discussions and opportunities to share ideas in a friendly and supportive environment. I am also indebted to my industrial partners for enabling this work to be performed - particularly Paul Packard and Hillary Sillitto.

On a personal note, it would be wrong not to acknowledge Ali Khajeh-Hosseini, Lakshitha de Silva and Dulani Meedeniya for being first class office mates and providing laughter, entertainment and silence during times when they were needed the most. Finally I thank Francina Hollis, now Francina Greenwood, and my parents for their enduring patience, love and support.

Abstract

Society is challenging systems engineers by demanding increasingly complex and integrated IT systems (Northrop et al., 2006; RAE, 2004) e.g. integrated enterprise resource planning systems, integrated healthcare systems and business critical services provisioned using cloud based resources. These types of IT system are often systems-of-systems (SoS). That is to say they are composed of multiple systems that are operated and managed by independent parties and are distributed across multiple organisational boundaries, geographies or legal jurisdictions (Maier, 1998).

SoS are notorious for becoming problematic due to interconnected technical and social issues. Practitioners claim that they are ill equipped to deal with the socio-technical challenges posed by system-of-systems. One of these challenges is to identify the socio-technical threats associated with building, operating and managing systems whose parts are distributed across organisational boundaries. Another is how-to troubleshoot these systems when they exhibit undesirable behaviour.

This thesis aims to provide a *modelling abstraction* and an extensible technique that enables practitioners to identify socio-technical threats prior to implementation and troubleshoot SoS post-implementation. This thesis evaluates existing modelling abstractions for their suitability to represent SoS and suggests that an agent-responsibility based modelling abstraction may provide a *practical* and *scalable* way of representing SoS for socio-technical threat identification and troubleshooting. The *practicality* and *scalability* of the abstraction is explored through the use of case studies that motivate the extension of existing responsibility-based techniques so that new classes of system (coalitions-of-systems) and new classes of threat (agent-related threats) may be analysed.

This thesis concludes that the notion of 'responsibility' is a promising abstraction for representing and analysing systems that are composed of parts that are independently managed and maintained by agents spanning multiple organisational boundaries e.g. systems-of-systems, enterprise-scale systems.

Contents

1. Thesis Overview

1.1 Introduction

Society is challenging systems engineers by demanding increasingly complex and integrated IT systems (Northrop et al., 2006; RAE, 2004) e.g. integrated enterprise resource planning systems, integrated healthcare systems and business critical services provisioned using cloud based resources. These types of IT system are often systems-of-systems (SoS). That is to say they are composed of multiple systems that are operated and managed by independent parties and are distributed across multiple organisational boundaries or legal jurisdictions (Maier, 1998).

SoS are notorious for becoming problematic due to interconnected technical and social issues. An exemplar is the NHS 'Care Record Service' - a failing attempt to integrate patient care records across NHS organisations in England. When the 'Care Record Service' contracts were awarded in 2003/4 it was intended that a detailed and integrated patient care record system would be delivered to all NHS trusts and GP practices by 2010. In 2009 the project was 4-5 years behind schedule, Accenture and Fujitsu – two of the four suppliers - had pulled out due to their inability to deliver, and the other suppliers, BT and CSC, had to revise their deliver schedules having recognised that more customisation was required than originally envisaged to meet each individual NHS organisations' needs (Committee, 2009; Health, 2011; NAO, 2011; Office, 2011). Analyses of this project suggest that it is suffering from socio-technical issues (Currie & Guah, 2007).

The above example suggests, and practitioners claim, that they are ill equipped to deal with the socio-technical challenges posed by system-of-systems. One of these challenges is to identify the socio-technical risks associated with building, operating and managing systems whose parts are distributed across organisational boundaries. Another is how-to troubleshoot these systems when they exhibit undesirable behaviour.

This thesis aims to provide a *modelling abstraction* and an extensible technique that enables practitioners to identify socio-technical threats and troubleshoot SoS post-implementation. This thesis evaluates existing modelling abstractions for their suitability to represent SoS and suggests that agent-responsibility based modelling abstractions may provide a *practical* and *scalable* way of representing SoS for socio-technical threat identification and troubleshooting. The *practicality* and *scalability* of the abstraction is explored through the use of case studies that motivate the extension of existing responsibility-based techniques so that new classes of system (coalitions-of-systems) and new classes of threat (agent-related threats) may be analysed.

1.2 Academic Merit

System-of-systems are worthy of academic study as many of the challenges these systems pose are relevant to the engineering of large-scale IT systems. This is especially the case *today* as we live in an era where:

- commercial pressures are driving the acquisition of ever more interconnected and sophisticated systems;
- the proliferation of technologies such as cloud computing create dependencies across multiple organisations with often differing commercial interests;
- consortia build, operate and maintain systems whose ongoing dependability is reliant on a 'coalition' of partners fulfilling obligations.

Perhaps the following five challenges are indicative of the academic and industrial significance of this research area:

- 'How do we identify and understand threats to the dependability of a SoS when it is dependent on the actions of multiple technical and non-technical agents operating in multiple organisations, whom may have differing self-interests or reside in different jurisdictions?'
- 'What modelling abstractions are suitable for representing these situations in a meaningful way to key stakeholders?'
- 'How do we take into account groups of human agent's abilities to resist/conflict with organisational changes when assessing the risks of deploying and operating a particular socio-technical configuration of an IT system?'
- 'What technical or non-technical structures can we put in place to facilitate the deployment and adoption of a large-scale IT system thus reducing risk?'
- 'How do we investigate and troubleshoot a large-scale IT system when it may be influenced by interactions between *technical* and *non-technical* elements, and where each stakeholder has a *partial view* of the situation?'

1.3 Problem Space

This thesis is specifically concerned with modelling SoS in order to ameliorate troublesome behaviour (troubleshooting) or identify potentially troublesome behaviour prior to its occurrence (threat identification).

When modelling a SoS, it is insufficient to simply represent its technical parts. Instead a SoS must be analysed as a *socio-technical system*. This is because to understand the structure and behaviour of a SoS, factors that affect the cooperation of humans interacting across organisational boundaries must be understood. These factors are not just technical since there is a need to identify and analyse problems that can arise due to interactions between groups or organisations that hold different commercial interests and perhaps operate in accordance to different laws, norms or domain standards.

Because of this need to understand interactions between socio-technical elements that span organisational boundaries, *SoS may be seen as a difficult class of socio-technical systems to model*. This is because they tend to be challenging to describe and understand due to their under-specification and their low comprehensibility. SoS are under-specified because:

- their specific operating conditions and exact configuration are too vast or ephemeral to completely specify (Hollnagel, 2008).
- their parts and interactions change over time and thus creating/maintaining complete structural and behavioural descriptions is too costly/difficult (Hollnagel, 2008).
- the number and nature of parts and interactions may be disputed, or no one person has a complete view of the system, and so multiple potentially inconsistent views of the system may exist.

SoS have a low comprehensibility because:

- interactions between system parts may be non-linear and emergent (Hollnagel, 2008)
- no individual stakeholder has a complete view of the system and its operating conditions
- stakeholders may need to make sense of a large number of parts, interactions and views to understand the system's structure and behaviour.

1.4 Solution Space

Modelling abstractions can help address the challenges of understanding SoS. They can help by minimising the effort or cost of describing and analysing them. A key challenge is to create modelling abstractions that provide the level of understanding required whilst being economical with respect to the quantity of information that must be collected and kept up-to date. To achieve this, a strategy for eliciting and representing a minimal set of necessary and sufficient parts and interactions is required.

In this thesis we are concerned with understanding the behaviour of SoS in order to ameliorate troublesome behaviour (troubleshoot) or identify potentially troublesome behaviour prior to its occurrence (risk / threat identification). To do this we develop a way of specifying what constitutes troublesome behaviour from each stakeholder's perspective using the notion of responsibilities. We also develop a way of establishing the bounds of the behaviour of a system so that claims can be made as to whether it is possible, or likely, that the SoS will act in a troublesome manner. To do this we postulate and test the hypothesis that a system's structure (as represented by a responsibility model) constrains the kinds of behaviour a system exhibits.

1.5 Thesis & General Approach

This thesis argues that *the notion of 'responsibility' provides a suitable abstraction for the socio-technical analysis of SoS*. We argue that it facilitates the enquiry, representation and understanding of situations by teasing out potentially intricate dependencies between technical and non-technical agents, whilst providing a suitable language for representing and discussing obligations, liabilities and norms which are important for understanding socio-technical threats and troubleshooting SoS. Our argument is evidenced by means of qualitative case studies that illustrate how the responsibility abstraction was used, or extended, to facilitate the analysis of real world systems.

A case study approach was selected, as the plausibility of this thesis' findings are dependent upon the phenomena of study being representative of SoS. Representative situations can only be found in-situ due to their nature and therefore a case study approach was selected. A mainly qualitative approach was selected as this enabled the exploration of situations via means of interviews and informal dialogue. Interviews and informal dialogue were selected because they are particularly effective at understanding the richness of 'messy', intricate and subtle situations.

1.6 Aim and Objectives

The principal aim of this research was to develop a *modelling abstraction* and an extensible technique that enables SoS to be analysed for socio-technical threats and troubleshot post-implementation. The objectives of this research were to:

1. investigate how responsibilities can be used for understanding socio-technical situations;
2. carry out case studies to investigate the *practicality* and *scalability* of the agent-responsibility modelling abstraction.

For the purposes of this thesis, an abstraction is said to be *practical* if it can be used to troubleshoot problematic systems and if it can be used to identify troublesome interactions prior to their occurrence. In other words an abstraction is *practical* if it enables the identification of socio-technical interactions that contribute to troublesome behaviour or that threaten to cause troublesome behaviour. For the purposes of this thesis an abstraction is *scalable* if it can be used for the practical analysis of SoS with non-trivial numbers of stakeholders and interacting components. That is to say that an abstraction can be used to troubleshoot and identify threats in systems that are composed of multiple systems that are operated and managed by independent parties and are distributed across organisational boundaries or legal jurisdictions.

This thesis claims that the 'responsibility' abstraction is *practical* as it provides ways of troubleshooting problematic socio-technical situations as well as ways of identifying socio-technical threats. This is illustrated through case studies that show that responsibility modelling can be used to:

- identify socio-technical interactions contributing to problematic system behaviour
- identify threats to system behaviour due to coalition partners holding incompatible commercial interests and so reneging on responsibilities
- identify threats to system deployment and operational effectiveness due to human agents' interests conflicting with those of the system being deployed/ in operation.

This thesis claims that the 'responsibility' abstraction may be *scalable*, as when combined with techniques from network analysis, situations with non-trivial numbers of elements may be analysable without an analyst having to manually inspect each and every interaction thus reducing the burden of human effort. This thesis demonstrates that metrics can be used to rank the importance of interactions and thus enable an analyst to understand a situation. This may therefore increase the scale of situation an analyst may analyse, or decrease the time required for an analyst to understand a situation. This use of metrics to rank elements is illustrated through a case study of a problematic enterprise document management system in a multinational systems engineering organisation.

1.7 Novel Contributions

This thesis provides four key contributions. Firstly, that *the notion of responsibility provides a suitable abstraction for assessing the socio-technical dependability of coalitions-of-systems* – that is to say, systems whose continuing operation are dependent upon a coalition of technical and organisational agents fulfilling obligations. This contribution is illustrated in chapter 5 via means of a cloud computing based case study and is also discussed in (Greenwood & Sommerville, 2011b).

The second contribution is that *the notion of 'responsibility' provides a suitable abstraction for identifying threats to system deployment and adoption situations where threats to human agents' cooperative behaviour need to be taken into account*. This contribution is illustrated in chapter 6 via means of a health sector case study and a cloud computing based case study.

The third contribution is that the notion of '*responsibility' provides a suitable component for a framework for troubleshooting enterprise-scale systems*. This contribution is illustrated in chapter 7 via means of a case study of a problematic enterprise document management system in a multinational systems engineering organisation and is also discussed in (Greenwood & Sommerville, 2011a).

The fourth contribution is that *responsibilities form part of a potentially scalable abstraction when used with techniques from network analysis*. This contribution is illustrated in chapter 8 via means of re-analysing the case study in chapter 7 to demonstrate that a problematic situation can be represented as a directed graph and that analytical techniques may be used as indicators of element 'importance' and 'complexity'. These techniques aid an analyst by indicating elements that may be important in situations where the number of nodes and their interconnections is to too large for a human to be able to analyze in a timely manner. This research is also discussed in (Greenwood & Sommerville, 2011c).

1.8 Thesis Structure

This thesis is structured so that chapter 2 reviews work related to system-of-systems (SoS) and their socio-technical analysis. The reader is introduced to the notion of system-of-systems and the problems faced when attempting to model SoS, followed by a survey of modelling paradigms and techniques from systems engineering and related disciplines. The survey considers approaches from systems engineering including functional decomposition based approaches (such as IDEF0 and FRAM) and agent-based approaches (such as responsibility modelling, I* and Tropos). Approaches from information systems and operations research are surveyed including Cultural Historical Activity Theory, Actor Network Theory, Rich Pictures and Cognitive maps.

Chapter 3 consists of a review of research methods suitable for the study of system-of-systems. It investigates the advantages and disadvantages of quantitative and qualitative data collection techniques, including surveys, interviews, participant observation, and data analysis approaches including multivariate statistics, network analysis and qualitative approaches such as grounded theory and hermeneutics.

Chapter 4 argues the practical and theoretical value of using responsibilities as an abstraction for the purpose of threat identification and troubleshooting. It is argued that responsibilities facilitate the enquiry, representation and understanding of situations by teasing out potentially intricate dependencies between technical and non-technical agents, whilst providing a suitable language for representing and discussing obligations, liabilities and norms which are important for understanding socio-technical threats and troubleshooting SoS.

Having established the practical and theoretical value of responsibility modelling, chapter 5 is concerned with extending responsibility modelling to enable the identification of socio-technical threats to the dependability of coalitions-of-systems. Coalitions-of-systems (CoS) are a sub-class of SoS that have the additional property that their subsystems interact to further overlapping self-interests rather than achieving an overarching mission/goal as in the case of typical SoS. Assessing the socio-technical dependability of CoS is an open research question of societal importance as existing socio-technical dependability analysis techniques typically do not assess threats associated with coalition partners reneging on responsibilities or leaving a coalition. We use a cloud computing based case study to demonstrate that a responsibility modelling based threat identification approach enables the identification of these threats. We provide first evidence that inspecting the distribution of liabilities among coalition partners may indicate the fragility of overlapping self-interests.

Chapter 6 is concerned with extending responsibility modelling to take into account threats to human agents' cooperative behaviour. Current approaches to threat analysis view stakeholders from a mechanistic means-end perspective where human agents are assumed to be passive and compliant. In this chapter, we use insights from conflict theory in social psychology to take into account factors that make stakeholders conflict with change. We make the case that this class of threat is an important class of threat that is missing from responsibility analysis. We used a cloud computing based case study to demonstrate that the conflict based threat identification approach enables the identification of this class of threats.

Chapter 7 is concerned with extending responsibility-based analysis to troubleshoot a problematic enterprise IT system. In this chapter we make the case that ethnographic and social analyses have not been widely assimilated into industry practice because they did not fit practitioner's practices. In response to this, we developed a lightweight qualitative approach to provide insights to ameliorate problematic system deployments. Unlike typical ethnographies and social analyses of *work activity* that inform systems *analysis* and *design*; we argue that analysis of *intentional* and *structural factors* to inform *system deployment* and *integration* can have a shorter time duration and yet can provide actionable insights. We evaluate our approach using a case study of a problematic enterprise document management system within a multinational systems engineering organisation. Our findings are of academic and practical significance as our approach demonstrates that structural-intentional analysis scales to enable the timely analysis of enterprise system deployments.

In chapter 8 we argue that techniques and tools for social network analysis can enable the analysis of problematic enterprise-scale socio-technical systems comprising large numbers of nodes. By means of reanalysing the case study in chapter 7, we demonstrate proof-of-concept tools for SoS scale network analysis and visualisation that may provide a promising avenue for identifying problematic elements and interactions amongst an overwhelming number of socio-technical elements. We demonstrate the potential of this approach by showing that: i) a problematic situation may be represented as a directed graph such that the elements in the situation are represented as nodes, and interactions between nodes as edges; ii) that eigenvector centrality may be used to rank the importance of elements in a situation and that highly ranked elements match those identified as important by a human analyst; iii) the 'complexity' of a situation, or a part of a situation, may be characterised using a feedback degree score which provides an indication of the extent elements are highly interconnected and involved in feedback loops. These findings indicate that computers may be used to aid the analysis of SoS situations by highlighting elements, or groups of interacting elements, that are important to the overall outcome of a problematic situation.

In chapter 9 we conclude that the notion of 'responsibility' is a promising abstraction for representing and analysing systems that are composed of parts that are independently managed and maintained by agents spanning multiple organisational boundaries e.g. systems-of-systems, coalitions of systems, enterprise-scale systems. We also set a future research agenda for the socio-technical analysis of systems from an engineering perspective.

2. System-of-Systems and their Socio-Technical Analysis

2.1 Introduction

The purpose of this chapter is to introduce the reader to research relevant to the socio-technical analysis of system-of-systems (SoS). The first section introduces the concept of a SoS and subsequent sections review the current state of socio-technical modelling and analysis abstractions and their applicability to SoS.

2.2 System-of-Systems

The study of 'system-of-systems' (SoS) is in a relatively embryonic form and as such the research community has not agreed upon a common definition of a SoS. The definitions that do exist are often domain dependent but tend to capture overlapping notions (Gorod, Sauser, & Boardman, 2008; Jamshidi, 2008; Pyster, 2011). That is to say, they capture the notion of two or more systems that are separately defined but operate together to fulfil a common goal (Checkland, 1999). They also typically postulate key characteristics such as: operational independence of component systems; managerial independence of component systems; geographical distribution; emergent behaviour; and evolutionary development processes (Maier, 1998). Other definitions of SoS include that of (Jamshidi, 2008) whom after reviewing six definitions of SoS favoured "systems of systems are large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal" (Jamshidi, 2008, p. 4).

This thesis defines a SoS as a class of system that delivers services or capabilities via the integration of independently managed and controlled systems. This definition is strongly influenced by (Maier, 1998) and is in line with the System Engineering Body of Knowledge (Pyster, 2011).

It is believed by the research community that SoS pose a set of distinct engineering challenges as the services or capabilities they provide cannot, by definition, be centrally managed because of their component systems' operational and managerial independence. It is believed that this independence creates a distributed control problem that needs to be understood throughout a SoS lifecycle. This independence presents challenges with respect to systems architecture, modelling and simulation, policymaking and standards (Jamshidi, 2008).

One important subset of these challenges are socio-technical (Jamshidi, 2008; McCarter & White, 2008). Socio-technical challenges arise since SoS are typically distributed across multiple organisational boundaries, geographies, or legal jurisdictions and their components often consist of individuals and organisational agents. An important aspect of this is to ensure components are not just integrated at a technical level but also at an organisational and legal level. Another important challenge is to ensure that the way components are integrated at the organisational and legal level does not inadvertently introduce risks that undermine functional or non-functional requirements. Another important challenge is how-to troubleshoot problems after system deployment. This is perceived to be particularly problematic since SoS are distributed across organisational boundaries and have independent management.

At present research in the area of the socio-technical analysis of systems of systems is in a nascent state. Due to a scarcity of SoS specific research, this thesis reviews approaches for representing and analysing socio-technical systems that are not necessarily SoS specific but assesses their potential applicability to SoS.

2.3 Paradigms for Representing and Analysing Socio-Technical Systems

In systems engineering, systems are often represented as models (Dickerson & Mavris, 2009). Models are representations of real/abstract things that are created for a purpose and only represent the details necessary and sufficient for that purpose (Anderson, Sweeny, Williams, 2000). Schematic models, the type of models this thesis focuses on, are conceptual representations that use notation to represent the entity being modelled.

Systems can often be usefully represented from multiple views. Four common views are:

- Structural views - representing the parts of a situation and their logical relations e.g. associations, aggregations, generalizations.
- Behavioural views - representing the parts of a situation and their causal interactions i.e. material or energy transfers or the transmission of data.
- Requirements views - specifying desired and undesired structural and behavioural properties of a system e.g. the system should not have a distributed architecture (structural property), the system should have the capability to process 1 billion transactions per second (behavioural property).
- Parametric views - specifying the critical engineering parameters of the system including those involved in evaluation of performance, reliability and physical characteristics. For instance, a performance characteristic of vehicle system may be $F \geq ma$, where m is its mass, a is its acceleration and F may be the maximum force produced by its braking system.

In choosing how-to model a system, practitioners select amongst modelling paradigms and abstractions. A modelling paradigm, in this context, consists of a set of simplifying assumptions that enable a system to be modelled. It has been suggested by (Melão & Pidd, 2000) that there are four basic modelling paradigms that may be used to model systems:

- the ‘machine metaphor’;
- the ‘organic metaphor’;
- the ‘feedback loop metaphor’;
- and the ‘socio-technical metaphor’.

These paradigms are explained in the following subsections. Abstractions will be discussed in section 4.

2.3.1 Machine Metaphor

The ‘machine metaphor’ treats a system as a deterministic[1] machine with a static structure that transforms inputs into outputs for some purpose. The metaphor is primarily concerned with representing a situation in terms of its static structure and factors that enable and constrain the system’s components to produce required outputs. It is assumed that components interact in a *linear*[2] manner and if any of the components are human beings they are treated as mechanisms that simply perform actions that contribute to converting inputs to outputs. Typical examples of modelling approaches that adopt the machine metaphor include data flow diagrams (Gane & Satson, 1979), IDEF0 models (NIST, 1993) and UML sequence, activity, state machine and use case models (Bennett, Skelton, & Lunn, 2005).

The machine metaphor has two significant limitations. The first limitation results from its treatment of human beings as mechanisms. As we know, human beings in the workplace do not simply perform actions that contribute to converting inputs to outputs. Human beings have basic needs and they seek to satisfy them in ways that do not contribute to converting inputs into outputs. For instance they may be satisfied through procrastination, developing relationships or being involved in organisational politics - so as to change the ‘system’ or resist undesirable changes to the ‘system’. The implication here is that the machine metaphor is unsuitable for analysing systems where human interests and politics need to be taken into account.

The second limitation of the machine metaphor results from its assumption that the system is static and its interactions are linear. This makes the machine metaphor unsuitable for understanding the behaviour of systems where components interact through feedback or their structure changes dynamically. This means that the machine metaphor is unsuitable for understanding systems that have non-linear[3] interactions between components or those that adapt

[1] The term deterministic indicates that there is no randomness in the future states of the system i.e. the system will always produce the same output for a given initial state.

[2] The term linear is used to denote that a system’s behaviour is not influenced by feedback loops i.e. the prior output of the system does not influence its future output.

[3] The term non-linear is used to indicate that a system’s behaviour is influenced by feedback loops i.e. the prior output of the system influences its future output.

structurally to internal dynamics or *environmental* changes. In response to these limitations three other paradigms have emerged that attempt to rectify these issues. These paradigms are the ‘organic metaphor’, which attempts to take into account adaptation to the environment, the ‘feedback loop metaphor’, which attempts to take into account non-linear interactions and the ‘socio-technical metaphor’, which attempts to take into account human/group needs and interests.

2.3.2 Organic Metaphor

The ‘organic metaphor’ extends the machine metaphor by treating a system as a deterministic machine that monitors and adapts to its environment in-order to survive. The organic metaphor adds to the machine metaphor by stating that the machine has a need for environmental resources or conditions to survive, and thus will reconfigure its structure, and thus behaviour, to best serve those needs.

To model this behaviour the organic metaphor introduces the notion of a boundary to supplement the mechanistic notions of input, transformation and output. The notion of boundary is used to indicate what is within the system, and thus may be restructured, and what is outside, thus acting as an environmental stimulus. This enables the organic metaphor to cope with situations where the system adapts to external factors. This can be useful when attempting to understand the effect of an increase in the cost of resources to operate a system, the effect of a system’s output on its own operation, or the effect of multiple systems sharing the same environment. According to (Melão & Pidd, 2000) discrete event modelling and simulation approaches such as SIMAN (de Vreede & van Eijck, 1998) form this paradigm. For further details on discrete event simulation see (Allen, 2011; Altiok & Melamed, 2007).

The limitations of the organic metaphor are that it does not recognise that the system may adapt due to internal factors (e.g. due to the humans’ socio-political actions) and that the interactions between parts may be *non-linear* e.g. that positive or negative feedback between components may occur. In order to cope with feedback dynamics the feedback loop metaphor is widely used. In order to cope with socio-political aspects a socio-technical metaphor is used.

2.3.3 Feedback loop metaphor

The ‘feedback loop metaphor’ (Forrester, 1958, 1971) treats a situation to be studied as a non-linear system. The system has a static structure that performs transformations and has a boundary from which inputs and resources are received, and outputs are deposited. The system is modelled as a flow of multiple inputs being transformed at various rates into various intermediate products until they are transformed into the system’s outputs. The non-linear interactions between parts may be understood by using mathematical models or simulations to derive an understanding of the flow of inputs, resources, intermediate products and the output. Typical examples of modelling approaches that adopt this paradigm include causal loop and stocks & flows approaches – for further information see (Sterman, 2000).

The limitations of the feedback loop metaphor are that it treats human aspects in a shallow manner as it views humans as input-output mechanisms to be controlled and to exercise control. This does not do justice to the fact that humans experience a richer socio-political existence, which is motivated by their individual and group interests. Another limitation of the feedback loop metaphor is that it assumes that the structure of a system is static. This means the metaphor is unsuitable for modelling complex adaptive systems however paradigms such as agent-based simulation may enable their study – for further information see (Epstein, 2007; Miller & Page, 2007).

2.3.4 Socio-Technical Metaphor

The socio-technical metaphor (Trist, 1981) contrasts with the previous paradigms by rejecting the notion of representing human beings as simple input-output mechanisms. The socio-technical metaphor recognises that human beings have needs and interests that may not be satisfied by simply contributing to the conversion of system inputs to outputs. The socio-technical metaphor encourages the analysis of these human needs so as to understand to what extent the system satisfies them.

A variety of differing approaches exist for doing this. Some focus on a human beings' fit with the tasks they are required to perform (Avison, Wood-Harper, Vidgen, & Wood, 1998; Mumford, 1995). Others focus on the fit between group level phenomena and the system (Checkland, 1999; Kaptelinin & Nardi, 2006) - for instance the implications of a system on departmental power, politics and general working culture.

The limitations of the socio-technical metaphor are that it assumes that sufficient information on human beings and the groups they form can be elicited, thus enabling useful analysis to be performed. In practice the gathering of information on humans, their thoughts and beliefs can be particularly challenging, especially at a large-scale. This will be discussed further in chapter 3, which focuses on data collection and analysis.

2.4 Modelling Abstractions for representing Socio-Technical Systems

There are a number of modelling abstractions (functional decomposition, agent-oriented decomposition, object-oriented decomposition ...) that enable the decomposition of a system in differing ways. Since the aim of this thesis is to provide an abstraction to enable analysts to represent and analyse SoS for the purpose of socio-technical threat identification and troubleshooting, this chapter largely restricts itself to abstractions that have been used for structural or behavioural modelling of socio-technical systems.

2.4.1 Functional Decomposition

Functional decomposition in its simplest form provides the notion of functions and functional relationships to decompose a situation into constituent parts (NIST, 1993). A function is a representation of an "activity, process, or transformation … identified by a verb or verb phrase that describes what must be accomplished" (NIST, 1993, p. 4). Each function has inputs, produces some output and is typically mediated by some controls and mechanisms. A mechanism represents the means by which a transformation is made and a control represents something that guides the mechanism. For instance, when representing the activity of 'system design' as a function, the mechanism would be a design engineer, the control would be design requirements, the input would be preliminary design data, and the output a detailed design.

Sophisticated functional modelling abstractions have been developed to enable the representation of elaborate systems from multiple perspectives using concepts such as viewpoints and hierarchical diagrams so that increasingly detailed information may be hidden and revealed when required. Once such approach is IDEF0 (NIST, 1993). IDEF0 uses the concept of parent-child diagrams in combination with the notion of purposes, viewpoints and environmental context to deliver this capability. Using IDEF0, a system is initially represented as a single 'top-level' function that receives inputs from and outputs to an explicitly described environmental context. A feature of the functional paradigm is that every function (including the 'top-level' function) can be further elaborated using child diagrams that represent the decomposition as a combination of interconnected sub-functions. This enables the decomposition to be performed numerous times until the level of detail obtained is judged to be sufficient for the purpose of the analysis.

IDEF0 has been used for the analysis of socio-technical systems although it has been used in a mechanistic manner. For instance (Imran, Foping, Feehan, & Dokas, 2010) used IDEF0 to model a water treatment plant (WTP) and found the notions of controls and mechanisms to be useful ways of identifying social aspects. They used these notions to represent relevant legislation and standards, capital funding, and human resources. They also found that the hierarchical decomposition that IDEF0 offers encouraged the identification of factors at different hierarchical levels.

Whilst (Imran et al., 2010) found IDEF0 to be useful, they noted its treatment of the social was limited because the approach did not allow for social factors to influence each other. They also noted that the IDEF notation is difficult for non-domain experts to understand. Their use of IDEF0 can also be further criticised, as although they claim to adopt a socio-technical approach, they do not consider the fit between human needs and tasks, nor between the technical input-output system and individual and departmental interests.

IDEF0 has been used and extended by (Romero, Company, Agost, & Vila, 2008) to represent an inter-organisational collaborative design system between organisations as a socio-technical system. Their extension, IDEF0+, extended the representation of functions to emphasise human collaboration (coordination and cooperation) aspects. They found that by distinguishing coordination and cooperation ICOM's (inputs, controls, outputs and mechanisms) from production ICOM's made their interpretation of diagrams easier. Again this use of IDEF0 was mechanistic, as non-linear interactions were not considered, nor was the fit between human needs and their activities, nor between the technical input-output system and individual/departmental interests

A similar critique is provided of the use of IDEF0 by (Pons & Raine, 2005) to model system design as a socio-technical system. Although IDEF0 was found to be useful as it assisted in identifying constraints on the design process and the sources of those constraints. For instance customer values and resources, market pressure, shareholder needs and so on. It did not go on to analyse factors that affect the fit between humans and their tasks, nor factors such as departmental power and politics that may influence the design process.

In a rare instance, IDEF0 has also been used to perform socio-technical analysis. It was used by (Waring & Wainwright, 2002) to model the interfaces between an NHS hospital's departments and facilitate communication in a highly political landscape. The processes in departments were represented as functions along with the information and resource flows between departments. Through this form of process modelling they found that participants were able to identify where power and politics was preventing new working practices. This was achieved by observing which departments and parties were controlling certain aspects of processes. The revelation of who had control over what, and for what purpose, opened opportunities for negotiation amongst staff to find improved states of affairs. In this study IDEF0 was successfully used to promote communication and understanding between departments by using IDEF0 as "means of a common language"(Waring & Wainwright, 2002, p. 408). IDEF0 was successfully used to focus on agreements, develop a shared understanding of customary behaviour, and maintain and develop social relationships.

IDEF0 suffers from a number of drawbacks that affect its suitability for modelling SoS situations. It is unclear whether IDEF0 can be used to model worker-task fit and also the politics or group processes present in a system. Waring & Wainwright (2002) demonstrated that IDEF0 models may be used to infer politics but they did not extend the notation to depict the politics within the IDEF0 notation. It is possible that IDEF0 may facilitate the analysis of worker-task fit as other functional approaches support this e.g. FRAM (Hollnagel, 2004; Hollnagel & Goteman, 2004). It is possible that politics or group processes may also be represented as a source of control for a function or a function itself. Further research in this area would help clarify this issue.

Within the resilience engineering community, functional decomposition has been demonstrated to be a useful abstraction for analysing and representing socio-technical systems. Functional resonance analysis method (FRAM) (Hollnagel, 2004; Hollnagel & Goteman, 2004), also known as functional resonance accident model, uses functional decomposition to describe how systems and environmental conditions can combine to produce desirable or problematic outcomes. FRAM uses a theory based on four principles to explain the success or failure of a system to transform its inputs into outputs (Hollnagel, Pruchnicki, Woltjer, & Etcher, 2008). These principles are:

- *The principle of equivalence of successes and failures* – the mechanisms of adaptation that enable a system to cope with complexity and thus be successful are also the mechanisms that enable a system to fail.
- *The principle of approximate adjustments* – the conditions that system operates in never completely match the conditions that have been specified or prescribed, so individuals and organisations must adapt/adjust the system to succeed in the current conditions.
- *The principle of emergence* – variability of normal performance is rarely sufficient to cause an accident or a malfunction. Typically, it is the variability of multiple functions that combine in unexpected ways. Both successful performance and failures are emergent as they cannot be attributed to individual parts acting on their own.
- *The principle of functional resonance* – the variability of functions can resonate causing the variability of another to exceed its normal limits. The consequences of this can then ripple through a system and result in a failure.

The FRAM approach is similar to IDEF0 in the sense that it has inputs, outputs, resources and controls. However FRAM differs as its functions have the explicit notion of temporal constraints. FRAM also differs from IDEF0 as it incorporates principles for understanding system behaviour.

FRAM also prescribes a number of steps to undertake to complete an analysis:

1. the essential system functions are identified;
2. the variability of these functions is characterised;
3. the analyst identifies situations where the variability of these functions may ‘resonate’ (i.e. become self-reinforcing or otherwise and exceed levels of variability tolerated by the system) resulting in undesirable outcomes;
4. barriers to prevent undesirable variability are identified.

FRAM provides a compelling approach for representing and analysing SoS as it has been demonstrated to be a practical technique for identifying socio-technical threats to complicated socio-technical systems.

FRAM has been used to analyse automated air traffic management systems (Woltjer & Hollnagel, 2008), explain the cause of air crashes (Hollnagel et al.,

2008) and also financial crashes (Sundström & Hollnagel, 2008). In these studies FRAM supported the analysis of the human-task fit so as to understand what circumstances would result in humans not successfully contributing the conversion of inputs into outputs. For instance in (Hollnagel et al., 2008), it was identified that the pilot's and co-pilot's information needs were not satisfied due to incorrect signs, their checklist contained only a single challenge response question, and the co-pilot was kept extremely busy prior to take-off such that he was too overloaded to double check the pilots runway selection.

Whilst FRAM is useful, its analysis falls short of the socio-technical paradigm. Humans-task fit is only understood with respect to the successful conversion of inputs to outputs. FRAM analysis neglects the analysis of job satisfaction and also group level interactions such as politics. It therefore misses threats due to workers altering standard procedures in-order to make their work more satisfactory and ignores threats that may arise due to political manoeuvring. In short, FRAM's treatment of humans as mere input-output mechanisms limits its usefulness to analysing socio-technical systems where it may be assumed all humans are satisfied by their work and that politics does not influence the system.

FRAM suffers from a number of other drawbacks that affect its suitability for modelling SoS situations. Firstly, without modification the approach does not appear to be suited for representing situations where the transformations being performed are uncertain, disputed, problematic, or where different stakeholders have different views on the desirability of their outcome. This limitation may perhaps be overcome as IDEF0 enables the representation of differing views and so further research is required to establish whether or not an IDEF0 style approach could deliver this capability. A second limitation is that it is unclear whether FRAM could be used, or extended, to model the effect of politics or group processes. Further research in this area would help resolve whether such social factors are representable using functional decomposition. Further research is also required to ascertain whether FRAM can scale up to the analysis of SoS situations. This suggests that further research in the use of computer-based analytics to analyse FRAM models could be a useful contribution.

2.4.2 Agent Oriented Decomposition

Agent orientation (E. Yu, 2002) is an abstraction for modelling systems in terms of building blocks that have characteristics such as intentionality, autonomy and sociality. It represents a shift away from representing systems are functions or objects towards representing systems in terms of building blocks of social significance. Two notable ways of approaching agent orientation are agent goal decomposition and agent responsibility decomposition.

2.4.2.1 Agent-goal Decomposition

Agent goal decomposition as used by I* (E. Yu, Giorgini, Maiden, & Mylopoulos, 2011; E. Yu & Mylopoulos, 1994) provides the notion of actors, goals and dependencies as abstractions to decompose situations into constituent parts. In this paradigm the structure of a situation is represented as a set of 'intentional' actors that depend on other actors to achieve a set of explicit goals and soft-goals. The term 'intentional' is used to claim that the actors being modelled have motivations, intentions and rationales that shape their actions (E. Yu et al., 2011).

The agent goal decomposition approach as used in I* is primarily a form of structural modelling - although it enables the creation of more sophisticated models that relate behavioural and structural views. Simple I* models comprise actors interacting via four causal dependencies.

The four causal dependencies consist of resource dependencies, task dependencies, goal dependencies and soft goal dependencies.

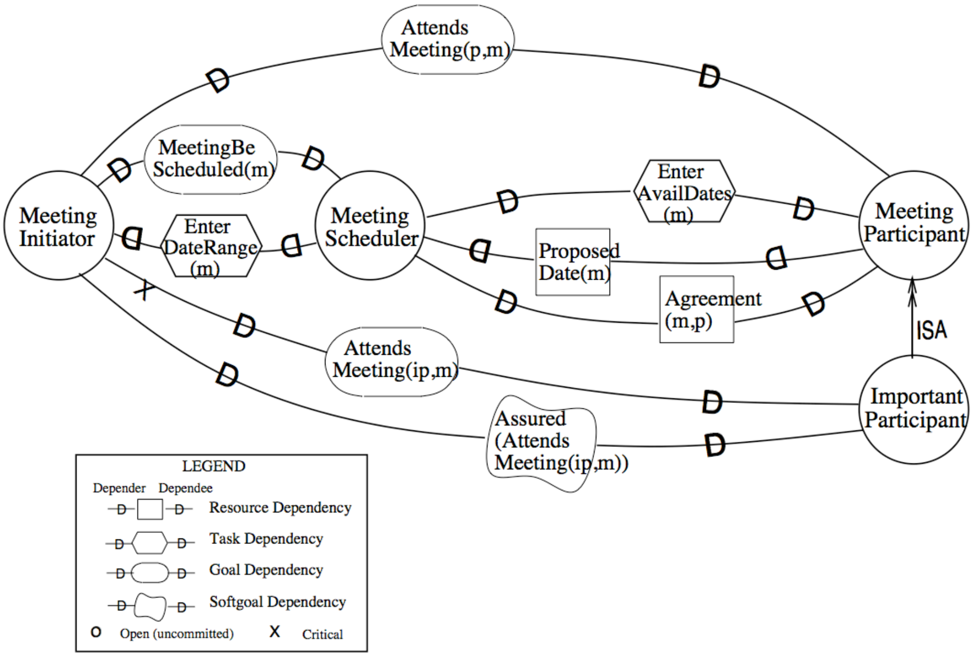


Figure 2.1 - Strategic Dependency model for a meeting (E. S. K. Yu, 1997)

A resource dependency illustrates that an actor's behaviour is causally dependent on another to provide a physical/informational entity (e.g. a desk, a worksheet). For instance in Figure 2.1 we can see that the meeting scheduler is dependent upon the information resource of an agreement on the meeting date, and that the meeting participants are dependent on the meeting scheduler proposing dates for the meeting.

A task dependency illustrates that an actor's behaviour causally depends on another to carry out a specified activity. For instance, the meeting scheduler is dependent on meeting participants entering dates that they are available. In a goal dependency an actor's behaviour is causally dependent on another to bring about a certain state in the world. For instance, in Figure 2.1 we can see the meeting initiator's goal of scheduling the meeting is dependent upon the meeting scheduler.

A soft goal dependency is where an actor's behaviour is causally dependent on another to perform some task that meets a soft goal. A soft goal requires context sensitive conditions to be met for the goal to be attained. For example the goal of "attending to patients needs promptly" is a soft goal since the manner in which the activity needs to be performed to meet the criteria is context specific as it may depend on the number of patients vs. staff, the seriousness of patients conditions, the experience of the staff and so on, such that it cannot be specified in a prescriptive manner. In Figure 2.1 we can see that the meeting initiator's softgoal of being assured that an important participant attends the meeting is dependent upon the important participant.

Agent goal decomposition has also been used to help reason about situations via the use of strategic rationale models. These models may be used to represent actors' reasoning about a situation using logical relations.

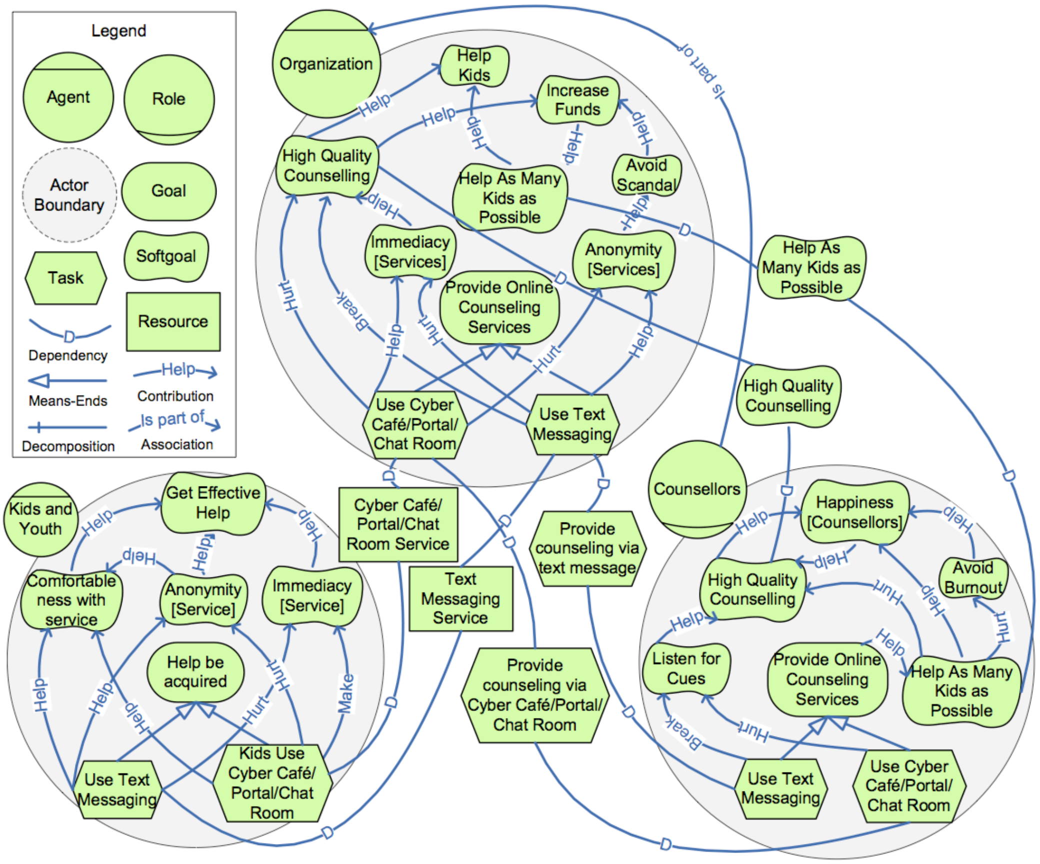


Figure 2.2 - A Strategic Rationale model for Youth Counselling Organisation (Horkoff & Yu, 2009)

Reasons are represented via the introduction of means-ends relationships (logical relations) that link the network of agent goal dependencies to actual or possible configurations of activity. This provides a means of analysing the ability of different configurations' to satisfy the agents' goals. Granular representations of situations may be constructed using decomposition links (logical relations), which enable a task to be decomposed into components such as sub goals, subtasks, resources for tasks and soft goals for tasks.

Even further granularity may be introduced by introducing concepts that distinguish between agents, roles and positions. In the language of I*, an agent is an actor with a physical manifestation such as a human being. A role is an abstract characterisation of a social actor within a specific context or domain e.g. Counsellor. A position is a set of roles typically played by one agent e.g. a person can simultaneously hold the roles of counsellor, tutor and doctoral student.

Agent goal decomposition provides an interesting candidate approach for representing and analysing SoS. It is a practical approach as it has been used to identify risks / threats. It has also been used to identify system configurations that meet actors' goal specifications. Within the domain of socio-technical systems, the I* framework has been shown to enable the identification of dependability threats and requirements for air traffic management (ATM) and enterprise systems (Maiden & Jones, 2004; Maiden, Kamdar, & Bush, 2006; Mayer, Dubois, & Rifaut, 2007). In the context of ATM this was achieved by means of exploring the consequences of agents fulfilling two or more roles and whether this affects the system's overall goal attainment (Maiden & Jones, 2004; Maiden et al., 2006).

In the context of enterprise systems, I* has been used to identify threats to security goals by modelling and discovering business assets, constraints and security requirements (Mayer et al., 2007). In these studies, human beings were treated as mere input-output conversion mechanisms and so human needs with respect job satisfaction and group level interactions such as politics were assumed to have no significant influence on the system. These assumptions limit these studies' relevance to the study of SoS as we may not assume that politics and worker satisfaction has no influence on the SoS.

The TROPOS Goal-Risk framework, a variant of I*, has also been used as an approach to analyse and mitigate threats to the goal accomplishment of an ATM system and a manufacturing organization (Asnar & Giorgini, 2007; Asnar, Moretti, Sebastianis, & Zannone, 2008). The TROPOS framework consists of a goal layer representing actor goals that should be obtained, an event layer that represents potential threats to goals, and a treatment layer that may be used to identify/represent possible threat management strategies. The systems under analysis were modelled as a configuration of related goals, tasks and events. This approach primarily focuses on the analysis of threats and the design of systems rather than the identification of threats. These studies also treated humans as mere input-output mechanisms and so the effects of politics and worker satisfaction were assumed to have no significant effect on the system.

An attractive aspect of the agent goal approach is that it supports scalability of analysis through the use of logics to support the satisfaction of actors' goals. It has been demonstrated that forward and backward propagation (Horkoff & Yu, 2009,

2010) enables analysts to infer whether a configuration meets actors' goals, identify configurations that do meet the actors' goals, and identify aspects of a configuration that result in the failure of goals. This characteristic of the agent goal approach is particularly promising for the analysis of SoS.

Agent goal decomposition has been used to analyse dependencies between actors to ensure the dependence is bidirectional and is thus 'sustainable' (Guizzardi, Perini, & Dignum, 2011). Agent goal decomposition has also been used to elicit and explain the dynamics of trust, confidentiality and distrust (Gans, Jarke, Kethers, Lakemeyer, & Schmitz, 2011).

However few studies have analysed the fit between human beings needs for job satisfaction and group level interests such as politics. One notable exception is the work of (Sutcliffe, 2011) that has extended agent goal modelling using complex adaptive systems theory to enable the analysis of power and motivation of actors in a situation. Despite this effort it is unclear whether this approach is scalable as the models produced require a subject matter expert to interpret and apply heuristics. It is also unclear to what extent the predictions of these models correlate with the observed behaviour of individuals or organisations.

More fundamentally it is unclear whether representing situations as interdependent *goal-seeking* actors is appropriate. The notion of a goal is a modelling construct rather than a notion that people commonly use to describe their working lives. Goals seem to be asocial constructs in that they merely represent states of affairs that an actor intends/desires to achieve, avoid or maintain. They provide little information about the organisational, social or cultural environment that may shape an actors behaviour, not what incentivises some behaviours over others. Perhaps it is unnecessary to capture and represent actor goals. Perhaps what is necessary is an understanding of constructs that shape the kinds of goals that actors pursue so that one may bound the kinds of behaviour actors will pursue without the cost of having to keep up-to-date models of what their goals are. This leads us to discuss agent responsibility decomposition.

2.4.2.2 Agent Responsibility Decomposition

Agent responsibility decomposition, as used in responsibility modelling (Dobson & Martin, 2007; Sommerville, 2007), provides the notion of responsible agents, responsibilities and dependencies to decompose a situation. This paradigm seeks to represent situations as a set of responsible agents, dependent on certain resources, seeking to fulfil their responsibilities. In this paradigm a responsibility is defined as a "duty, held by some agent, to achieve, maintain or avoid some given state, subject to conformance with organisational, social and cultural norms" (Lock, Storer, & Sommerville, 2009). The term duty is carefully defined to capture notions of obligation and accountability such that if an agent does not appropriately discharge their obligation they will be held liable. This modelling of obligations and liabilities attempts to tap into the social forces compelling agents to act in particular ways and thus is hoped to provide information about possible agent behaviours. The phrase conformance with organisational, social and cultural norms captures the fact that responsibilities must be discharged in accordance with legal and domain standards.

Agent responsibility decomposition is both a structural and behavioural modelling approach. Responsibilities represent units of behaviour that an agent has a duty to perform. Resources represent information or equipment that an agent requires to fulfil a responsibility. Hence a responsibility model is a description of the behaviours that agents have a duty to perform and the resources they require to perform these behaviours.

In its simplest form the relationships between agents, resources and responsibilities comprise the ‘responsible for’ relation that denotes that an agent is responsible for the performance of a duty, the ‘has’ relation that denotes the allocation of a resource to an agent or to a responsibility – see Figure 2.3 below.

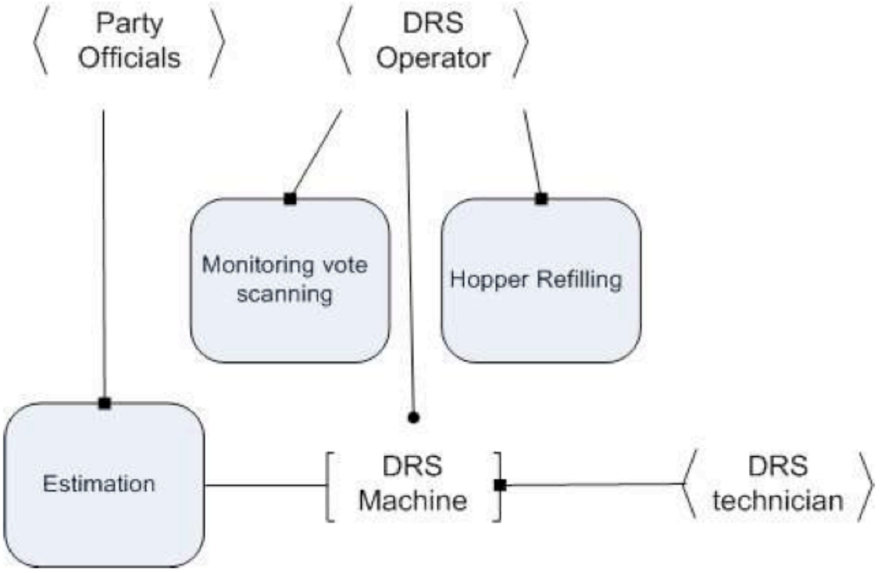


Figure 2.3 - Responsibility Model of e-Counting System in Scottish Elections (Lock, 2009)

In more complicated uses of the approach, relations such as: the ‘subordinates to’ are used to indicate that one agent is under the authority and control of another; the ‘assignment’ relation to indicate that one agent has assigned a responsibility to another; the ‘membership’ relation to indicate that an agent, responsibility or resource is also part of another e.g. the police service, ambulance service and fire service are parts of ‘Silver command’ - a crisis response organisation – see Figure 2.4 below.

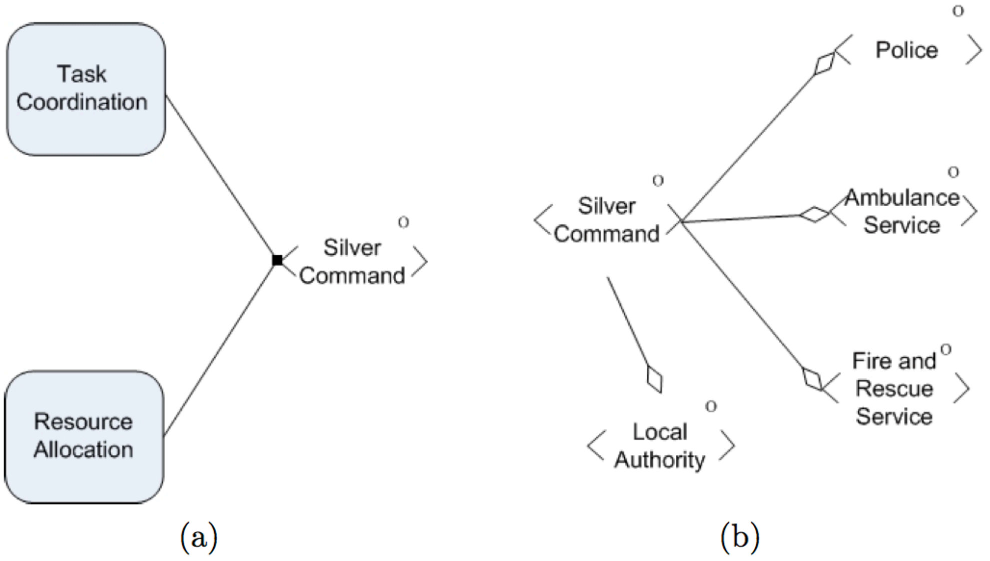


Figure 2.4 - Responsibility Model of Responsibilities during a Civil Emergency (Sommerville, 2009)

The agent responsibility paradigm can be seen as an alternative to the agent *goal* paradigm as it takes the stance that situations can be understood by understanding factors that provide a social / legal bound on the behaviour of agents, such as obligations and liabilities, rather than framing the agents' behaviours in terms of explicit goals or soft goals.

The agent responsibility approach has been successfully used by (Lock et al., 2009; Sommerville, Storer, & Lock, 2009) to analyse the failure of an E-counting system in the Scottish elections and to analyse UK civil emergency plans. In the E-counting case, agent responsibility decomposition was used to represent the dependencies between human/organisational agents and resources that were required for the socio-technical e-counting system to operate successfully. The identification and analysis of failures was achieved through the use of a HAZOPS style approach. This comprised considering the implications of hazard keywords on each dependence and identify threats that could emerge should agents renege on their responsibilities, resources fail, or become available too early, too late or in insufficient number (Lock et al., 2009). When used to analyse civil emergency plans a similar HAZOPS approach was used along with a rich set of structural relations to describe the complicated relationships between agents and the assignment of responsibilities.

Agent responsibility decomposition is an interesting candidate for modelling SoS situations as it is a simple and practical approach. Its minimalist structural representations seem less effortful to elicit than task/process level approaches and can be used to identify threats and potentially for troubleshooting situations. The approach also promises to be relatively scalable as computer support has been developed that identifies weaknesses in models, incomplete models, and models where responsibilities have not been assigned to agents (Lock et al., 2009; Sommerville et al., 2009).

A drawback of agent responsibility decomposition is that its socio-technical analysis is weak. The agent responsibility approach treats human beings as mere mechanisms for fulfilling responsibilities. It does not consider the fit between assigned responsibilities and the human or organisational agent, or whether an agent would find a responsibility satisfying or otherwise. Equally the approach does not take into account politics, group processes and other factors that can arise when responsibilities are assigned and distributed amongst differing groups. More research is required in this area to determine how these 'social factors' could be incorporated into responsibility modelling.

Another drawback of agent responsibility decomposition is that, to date, analyses have assumed that interactions between system parts will not exhibit non-linear properties for instance triggering cascades or forming feedback loops. This limitation is significant as in reality threats can trigger a sequence of events that ripple through a system. More research is required so that the analysis of non-linear behaviour may be incorporated to agent responsibility decomposition approaches. Further research may also be required in exploring how logics or other analytical techniques could be used to improve support for computer-aided analysis of these models.

2.4.3 Information Systems

Insights from the discipline of 'Information Systems' (IS) may be relevant to the socio-technical analysis of system-of-systems this is because an 'information system' is "the system that emerges from the mutually transformational interactions between the information technology and the organization" (Lee, 2004, p. 11). This means that the discipline of information systems is interested in understanding the relationship between people and information technology at the organisational scale. This suggests that theories and approaches to understanding information systems may be applicable to the modelling and socio-technical analysis of system-of-systems. This section will cover two interesting approaches to the analysis of information systems – cultural-historical activity theory and actor-network theory.

2.4.3.1 Cultural-Historical Activity Theory

Cultural-historical activity theory (CHAT) is a theoretical framework for describing and understanding human activity within its broader socio-cultural context (Kaptelinin & Nardi, 2006). For the purposes of system engineering, CHAT may be thought of as a set of theories for understanding human behaviour at an individual and group level.

Compared to typical systems engineering approaches, CHAT re-orients analysis to understand how an information technology contributes to satisfying a human's, or an organisation's, needs rather than analysing how a human may contribute or hinder the conversion of inputs into outputs. This flipping of the orientation of analysis provides a human-centred analysis of systems.

Central to activity theory is the notion of activity. Activity is the purposeful interaction of a subject (an intentional agent such as a person or an organisation) with the world. Purposeful interaction is defined as a process by which mutual transformations occur between the 'subject' and 'object' – the 'object' of an activity is some prospective outcome that the human or organisation needs or desires.

According to CHAT, activities have hierarchical structures that are composed of three layers.

- *Motives* - The top most layer is the motive of an activity. A motive is a need or desire that motivates an activity.

- *Goals* - At the next layer, the action layer, a subject's goals are the object of analysis. Goals are conscious such that a subject is aware of their existence and that they structure their actions. This is in contrast to motives, which are subconscious unless actively elicited by an individual. For instance, in the mornings I am acutely aware that my goal is to get to a bus stop on time so I can get to work by 9:30am. Because of my awareness of this goal, I structure my actions such that the route I walk to the bus stop and the type of breakfast I prepare in the morning is compatible with attaining this goal.

- *Operations* - The next layer comprises the operations layer. This is where analysis is directed towards the conditions under which a subject is trying to attain a goal. Operations are routine processes (that require almost no conscious thought) that provide an adjustment to an action due to the ongoing circumstances. A common example of an operation is the avoidance of people whilst walking along a crowded high street. This requires little thought and occurs in the background whilst a person typically focuses on their goals, such as getting to their bus stop on time or posting their mother's birthday card.

In CHAT, a socio-technical system is cast as a mediator of the subject-object relationship. This means the socio-technical system is regarded as something that may facilitate, inhibit, or alter the subject's abilities to satisfy their various needs. This facilitation, inhibition or alteration is achieved by means of influencing a subject's operations, goals and motives. For instance the system may provide a means of achieving what was previously unachievable or may prevent previously achievable goals. The system may also have the ability to alter a subject's operations, goals or motives through a process termed development. Development is the idea that subjects alter their behaviours in response to their experiences. According to CHAT development is a socio-cultural process and therefore to understand the behaviour of subjects it requires an understanding of cultural practices that may influence their development.

CHAT provides a useful framework for troubleshooting problematic socio-technical systems by understanding the ways in which subjects motives are unsatisfied. CHAT may be used to analyse whether a subject's work activity and environment enables them to satisfy their motives.

When an organisation is the subject of study it is treated as socially distributed human activity and so the role of specialisation and the division of labour is given much attention. This is because socially distributed work activity results in actions that are motivated by one object (say the need for food and shelter) but are directed at another (say the goal of writing an excellent textbook).

In modern organisations and society, there can be complex relationships between objects of motivation and objects of action sometimes resulting in disconnections between motivating objects and objects of action, thus resulting in dysfunctional organisational behaviour. These situations can become further complicated by incompatibilities between the motives and goals of the organisation and those of the individuals whom are members of the organisation. Symptoms of this include bureaucracy reigning over common sense and local optimisations at the expense of the organization as a whole. CHAT enables the analysis of these inconsistencies within socio-technical systems.

The work of Yrjö Engeström (Engestrom, 1999, 2000) on activity systems provides a useful framework for analysing and troubleshooting socio-technical systems. Engeström has added to activity theory by introducing a third component to subject-object interactions, the notion of the community. This means that activity is defined as a process by which mutual transformations occur between the 'subject', 'object' and the 'community'. Engeström further elaborates this model by introducing three different types of mediator: tools, rules and the division of labour. These six elements enable the description of complex technology mediated social practices (e.g. a problematic situation) and enable the identification of incompatibilities between elements that may be resulting in dysfunctional situations – see Figure 2.5 which provides a schematic of the interactions that occur during human activity according to his activity systems perspective

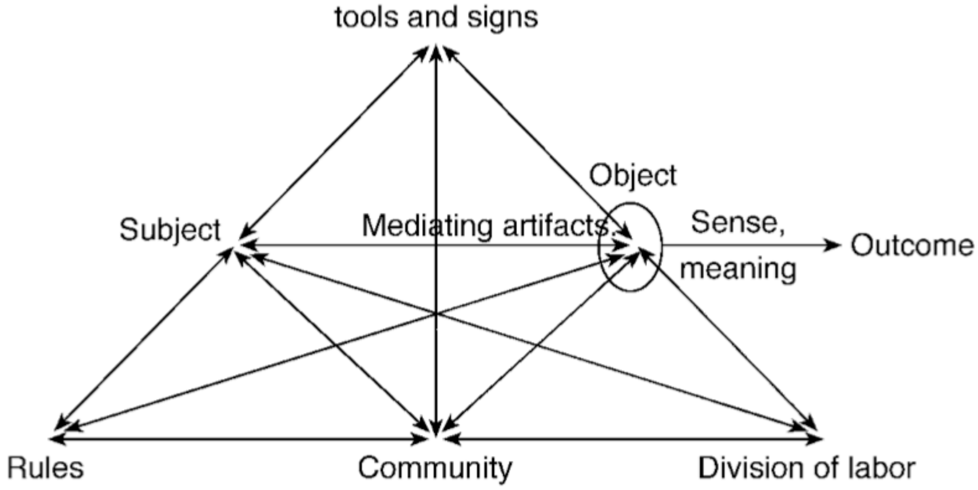


Figure 2.5 – The structure of human activity systems (Engestrom, 2001)

Figure 2.5 provides a schematic of the interactions that occur when a subject performs purposeful activity. It shows that the subject-object relationship is influenced by the tools and signs available in the situation, the formal/informal rules of the situation, the norms/expectations of the community that the subject is a part of, and the way work is divided up in the situation. It also shows that the subject-object relationship mediates the activity that is performed and the interpretation of the outcome of the activity. The diagram is considered to be useful as it highlights aspects of a situation an analyst may want to investigate to identify mediators that encourage behaviour that is inconsistent with a desirable outcome.

Engeström's framework has been used to analyse a variety of problematic socio-technical systems. For instance it has been used to support the study and organisational redesign of a children's hospital outpatient clinic (Engestrom, 2000, 2001), understand conflicting goals of judging in legal cases (Engestrom, 1996), studying the problematic nature of inter-organisational partnerships in the construction industry (Hartmann & Bresnen, 2011), and studying the use of technology in a higher education context to support learning (Isssroff & Scanlon, 2002).

In early use, the activity systems framework was limited to simple unitary views of systems. For instance in (Engestrom, 2000) the framework is used to analyse the sequence of actions that form the process of assessing an outpatient and deciding whether the patient requires inpatient care. This analysis was limited because it solely focused on disturbances / incompatibilities that inhibited the transformation of system inputs (sick patients) into system outputs (less sick patients). There was little systematic analysis of whether the needs of each human being involved in the system was being satisfied. The result was an analysis that focused on improving the manner in which system outputs (less sick patients) were produced without explicit consideration of satisfying physicians, nurses, managers and patients' needs. These socio-technical factors are only treated implicitly when hospital staff are brought together to negotiate new ways of working that minimise disturbances / incompatibilities.

In later work (Engestrom, 2001) pluralist views of systems were adopted. The notion of a 'boundary crossing laboratory' was developed so that the differing views, for instance of patients, health centre practitioners and hospital practitioners could be aired and discussed. This enabled the attending participants to generate a shared understanding of the interactions between their activity systems and their consequences for patient care. This particular approach is interesting as it uses the conflicting needs and views of participant to drive a resolution that satisfies or at least improves upon the current system.

The drawbacks of this approach are with respect to its scalability. The approach does not appear to be scalable to large situations since representatives from each viewpoint to be considered must be present and attend the boundary crossing laboratory meeting. Another drawback of the approach is that it relies on the participants identifying incompatibilities on the fly during the meeting. This can be troublesome in some situations as expertise and judgement is required to spot subtle incompatibilities between objects. Another limitation arises from the fact that incompatibilities are identified through discussion. This lack of more formal representation of interactions makes systematic analysis more challenging as it is difficult to track the effect of multiple interactions.

To address these limitations, the activity systems framework was expanded by Halloran (Halloran, 2000, 2001). His extension provides a visual way of representing incompatibilities between subjects' activity systems at both the individual and organisational level. This enables the analyst to bridge between the individual level and the organisational levels of analysis in a systematic manner – See Figure 2.6.

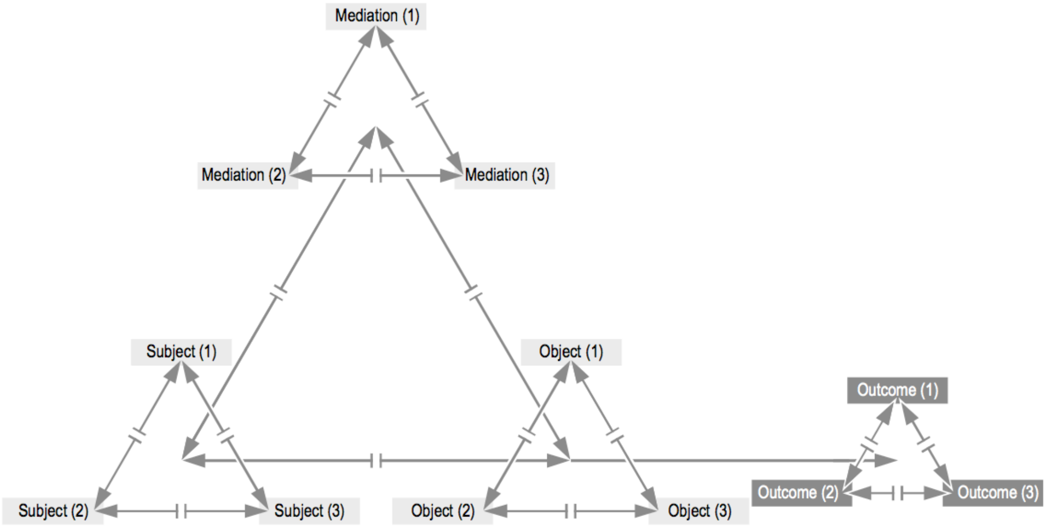


Figure 2.6 - Activity Space Framework (Halloran, 2000)

The activity space framework has been used to study the introduction of new technology in a radiology department (Halloran, 2000), and the use of groupware technology in the higher education (Halloran, 2001).

The CHAT approach is an interesting approach to modellers of SoS situations. It is practical in the sense that it may be used to troubleshoot problematic situations and may be used to identify potential issues if a technology is to be deployed. The approach also appears to scale when representing small to medium scale situations and there exists some computer support to enable this – see (Halloran, 2000). Another useful feature of CHAT is that it encourages the analysis of the 'social' by eliciting tensions between subjects, objects and mediators thus enabling the development of an understanding of the 'social' in terms of intervenable entities. Another interesting characteristic is its focus on learning and development. Successful interventions are those where the organisation learns to adapt to its situation and so theories that facilitate this are beneficial.

A key limitation of the CHAT approach is that its data collection and analysis approach would be extremely arduous for studying large-scale systems. This is because its interpretivist approach relies upon a human analyst identifying important or influential sets of interactions / inconsistencies in a situation. There are no case studies to my knowledge indicating that analytical techniques may be used to support an analyst in understanding situations that are too large or complicated for a human being to keep track of the many interactions. Another limitation is that CHAT is typically used to guide and interpret ethnographic studies and therefore this style of enquiry may be too time consuming should it be used to systematically analyse a SoS situation.

2.4.3.2 Actor Network Theory

Actor network theory (ANT) is a school of thought that attempts to understand the 'social' in terms of networks of associations. ANT attempts to expose what the 'social' *is* by means of tracing associations between human and non-human actors (Latour, 2005). By analysing 'assemblages' of associations ANT aims to provide explanations of how the 'social' is constructed. This is in direct contrast to other forms of sociology where 'social phenomena' are often explained in terms of other 'social phenomena' - for instance poverty and inequality being explained in terms of asymmetries of social power. Instead ANT seeks to decompose the 'social' in terms of anything that can be seen to be *acting* in a situation (Latour, 2005) - for instance explaining poverty and inequality in terms of geographical resources, scarcity of reliable transportation links and so on.

For the purposes of systems engineering, ANT provides a way of representing and analysing socio-technical systems without first dividing the focus of enquiry into 'socio' and 'technical' parts and then attempting to understand their 'interface'. Instead it tries to account for systems by describing:

- how they emerged from a number of associations among heterogeneous human and non-human actors spread out across time and geographical space;
- how the network of human and non-human actors resulted in some sort of stable or otherwise configuration.

ANT is unlike typical systems paradigms as it avoids the notion of a system and rejects the notion that human and non-humans may be assumed to be input-output mechanisms[4]. Instead it advocates that there are networks of interacting human and non-human actors that have their own ideas about what is going on, how it's going on and why they behave in a certain way. In some ways ANT may be thought of as being a radical example of the socio-technical metaphor as it attempts to understand the needs and perspectives of non-human actors in addition to human actors.

The reason why ANT attempts to understand the needs and perspectives of non-human and human actors is that ANT does not want to preclude anything from being an actor *a priori*. Another reason is that human designed objects can embody their designers perspectives or promote certain social or cultural values (Lash, 2001; Winner, 1997). Therefore ANT permits an actor to be "*any thing* that does modify a state of affairs by making a difference" (Latour, 2005, p. 71). This means material objects such as clothes, hammers, and speed bumps count as actors if their presence in a situation modifies/mediates the state of affairs. For instance, walking along a high street with/without clothes changes the state of affairs considerably. Similarly ANT also does not preclude any types of association, for instance, actors may be associated by authorize, enable, allow, forbid, make possible, block, and so on. By building up networks of these associations between actors one hopes to describe the phenomena of interest in an unprejudiced manner.

To avoid privileging certain actors' perspectives ANT does not assume any particular ontology. Instead situations are represented using the actors' own concepts of reality. This means ANT focuses on actors' definitions and ideas of what, how and why things are going on. ANT strongly rejects the notion that analysts should translate the actors' worlds into a normalized scientific form made from a number of standardized concepts. Instead it encourages the understanding of situations by comparing and contrasting the subtleties between actors' incommensurable worldviews.

To understand a socio-technical system the analyst should seek to understand 'controversies'. Metaphorically, 'controversies' may be thought of as tensions that attempt to steer actors in different directions. In ANT, the most important 'controversies' to understand are those with respect to group formation and agency. The group formation controversy seeks to understand why actors are

[4] ANT does not claim *a priori* that systems do not exist. It simply claims that one may not assume that the phenomenon of study is a system. Whether something is a system is something to be investigated by tracing associations between actors.

arranged in one particular configuration rather any other. In practice this means identifying actors that make the present group boundaries the way they are, rather than some other configuration. This often comprises identifying human actors, such as spokespersons, and material actors, such as legal documents, that define what the group should be about and its history.

The agency controversy seeks to understand what agencies are at work in producing the situation. This question is important as it recognizes the intuition that in 'social' situations people are not free to do literally anything they want, but instead play 'social roles' because of 'social forces' compelling a person to act in a 'socially acceptable' way. The aim here is to trace associations amongst actors to understand how these roles, forces and standards emerge. In practice this means gathering actors' accounts of their actions and understanding the influences as they perceive them.

Since some actors may be non-human, i.e. objects such as clothes, hammers and speed bumps, gathering actors' accounts of their actions may initially appear impossible - if not ludicrous. To workaround this snag, observational techniques have been invented to make objects offer non-verbal descriptions of themselves. These techniques consist of visiting locations where their behaviours and characteristics may be observed such as in innovation settings like engineers' workshops, marketers' focus groups and end-users' homes. In these locations an object's 'social life' is observable as it is in the process of being created and so associations between plans, regulations, standards and so on are visible. Other techniques may also be used to gather accounts - for instance:

- studying situations where the object is distanced from its usual location;
- observing the absence of an objects effects when it is broken;
- examining historical records of its development.

Situations where objects are distanced from their usual locations includes places where objects are distanced in *time* (such as in archaeology), distanced in *geography* so that it occupies a different 'socio-cultural space', or distanced in *skills* such that other actors need to learn how to associate with it. In all these locations, a process of learning, trial and error and innovation takes place and so it helps reveal a description of the non-human actors' needs and their view of the world.

ANT has been used in many case studies to understand problematic situations. For example, disputes surrounding the development of a web portal at the world bank (Marres, 2004), how the reliability of health information on the web is negotiated and attained (Adams & Berg, 2004), the contested evolution of web browser technology (Faraj, Kwon, & Watts, 2004) and IT project escalation (Mähring, Holmström, Keil, & Montealegre, 2004).

ANT is an interesting paradigm to modellers of SoS situations as its analysis of the 'social' is extremely thorough in the sense that it demands the creation of descriptions that are grounded in the observable interactions of human and non-human actors. This is particularly attractive if one is trying to troubleshoot a problematic situation as the descriptions provided explain how the situation is assembled and maintained by the behaviour of people and materials rather than intangible constructs.

The downside of ANT style analyses are that they are typically time intensive since they comprise a meticulous form of ethnography and so further research is required to investigate whether these accounts can be generated in a timely manner by using ANT style analyses sparingly in particularly critical parts of a situation. Another limitation of ANT style analyses are that they comprise dense textual accounts of a situation and therefore as a method of representation for articulating, analysing and understanding a situation they can be unwieldy and are unlikely to scale up to large-scale SoS situations. Further research would be required to develop a means of representing this information in a condensed form that could be analysed by analytic means thus supporting the SoS analyst.

2.4.5 Operations Research

Operations research (OR) is a discipline concerned with analysing complex problems and helping decision-makers work out the best means of achieving a desired outcome, or even figuring out what a desired outcome might be. Since the 1970s a sub-discipline of OR, called soft OR, has developed many techniques to support decision-making in ‘soft’, ‘messy’, ‘wicked’ or difficult to quantify situations. In this subsection we will review a selection of the pioneering work on abstractions for representing and analysing socio-technical situations that have emerged from this discipline.

2.4.5.1 Rich Pictures

The rich picture technique provides the notion of free form sketches as abstractions to decompose a situation into its constituent parts (Checkland, 1999). The paradigm does not presuppose a set of modelling abstractions but instead suggests that the analyst includes elements that are representations of ‘structure’, ‘process’ and ‘climate’. Structure is suggested to include the physical layout of a situation, the power hierarchy of people or groups, their reporting structures and patterns of formal and informal communication. Process is suggested to include what activities are performed, how decisions are made, how monitoring is done, and the external effects of these activities. Climate is suggested to include the relationship between the situation’s structure and the processes performed. This includes social and political aspects of a situation such as attitudes, beliefs, roles, norms or values as it is hypothesised that it is these that mediate the activities of the people and groups acting within the situations structure.

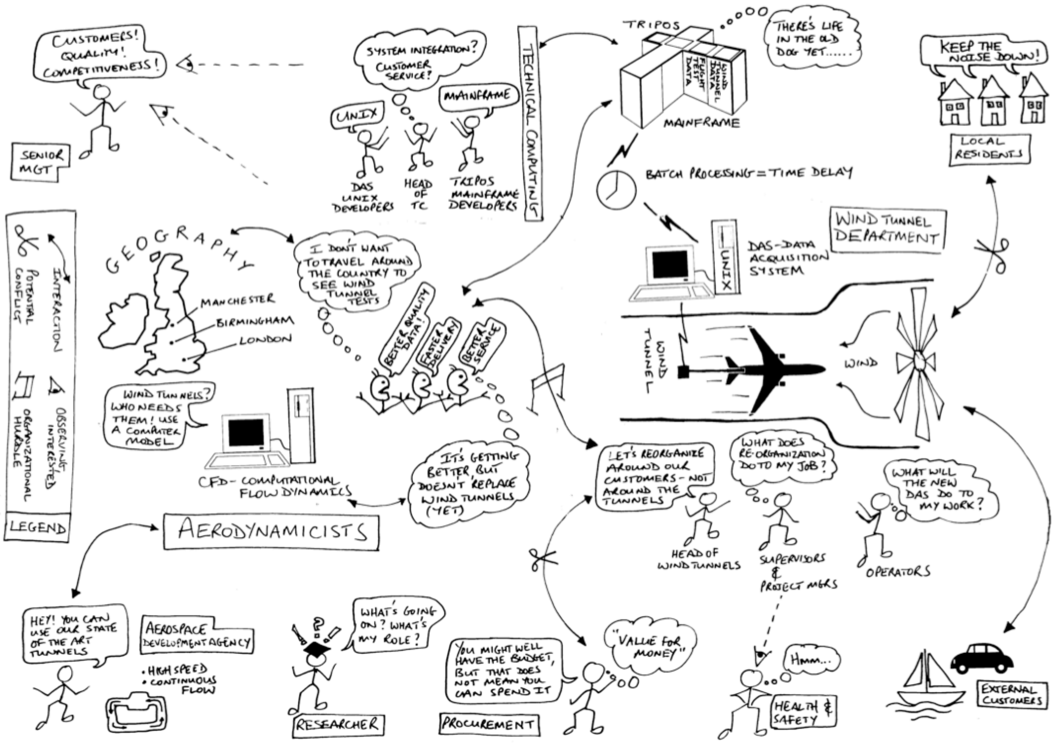

CUSTOMERS! QUALITY! COMPETITIVENESS!
SENIOR MGT
SYSTEM INTEGRATION? CUSTOMER SERVICE?
UNIX
MAINFRAME
DAS UNIX DEVELOPERS
HEAD OF TC
TRIPOS MAINFRAME DEVELOPERS
TECHNICAL COMPUTING
TRIPOS
FLIGHT TEST DATA
WIND TUNNEL DATA
MAINFRAME
THERE'S LIFE IN THE OLD DOG YET......
KEEP THE NOISE DOWN!
LOCAL RESIDENTS
BATCH PROCESSING = TIME DELAY
WIND TUNNEL DEPARTMENT
UNIX
DAS-DATA ACQUISITION SYSTEM
WIND TUNNEL
WIND
POTENTIAL CONFLICT
INTERACTION
ORGANIZATIONAL HURDLE
OBSERVING INTERESTED
LEGEND
GEOGRAPHY
MANCHESTER
BIRMINGHAM
LONDON
I DON'T WANT TO TRAVEL AROUND THE COUNTRY TO SEE WIND TUNNEL TESTS
BETTER QUALITY DATA!
FASTER DELIVERY
BETTER SERVICE
WIND TUNNELS? WHO NEEDS THEM! USE A COMPUTER MODEL
CFD - COMPUTATIONAL FLOW DYNAMICS
IT'S GETTING BETTER, BUT DOESN'T REPLACE WIND TUNNELS (YET)
AERODYNAMICISTS
LET'S REORGANIZE AROUND OUR CUSTOMERS - NOT AROUND THE TUNNELS
WHAT DOES RE-ORGANIZATION DO TO MY JOB?
WHAT WILL THE NEW DAS DO TO MY WORK?
HEAD OF WIND TUNNELS
SUPERVISORS & PROJECT MGRS
OPERATORS
HEY! YOU CAN USE OUR STATE OF THE ART TUNNELS
AEROSPACE DEVELOPMENT AGENCY
• HIGH SPEED
• CONTINUOUS FLOW
WHAT'S GOING ON? WHAT'S MY ROLE?
RESEARCHER
YOU MIGHT WELL HAVE THE BUDGET, BUT THAT DOES NOT MEAN YOU CAN SPEND IT
PROCUREMENT
"VALUE FOR MONEY"
Hmm...
HEALTH & SAFETY
EXTERNAL CUSTOMERS

Figure 2.7 - A Rich Picture reflecting the development of a wind tunnel (Vidgen, 1997)

Rich pictures have been used as part of soft systems methodology (SSM) on numerous projects that have consisted of understanding problematic socio-technical situations ranging from general problem solving through to organisational design (Mingers & Taylor, 1992). These projects have included:

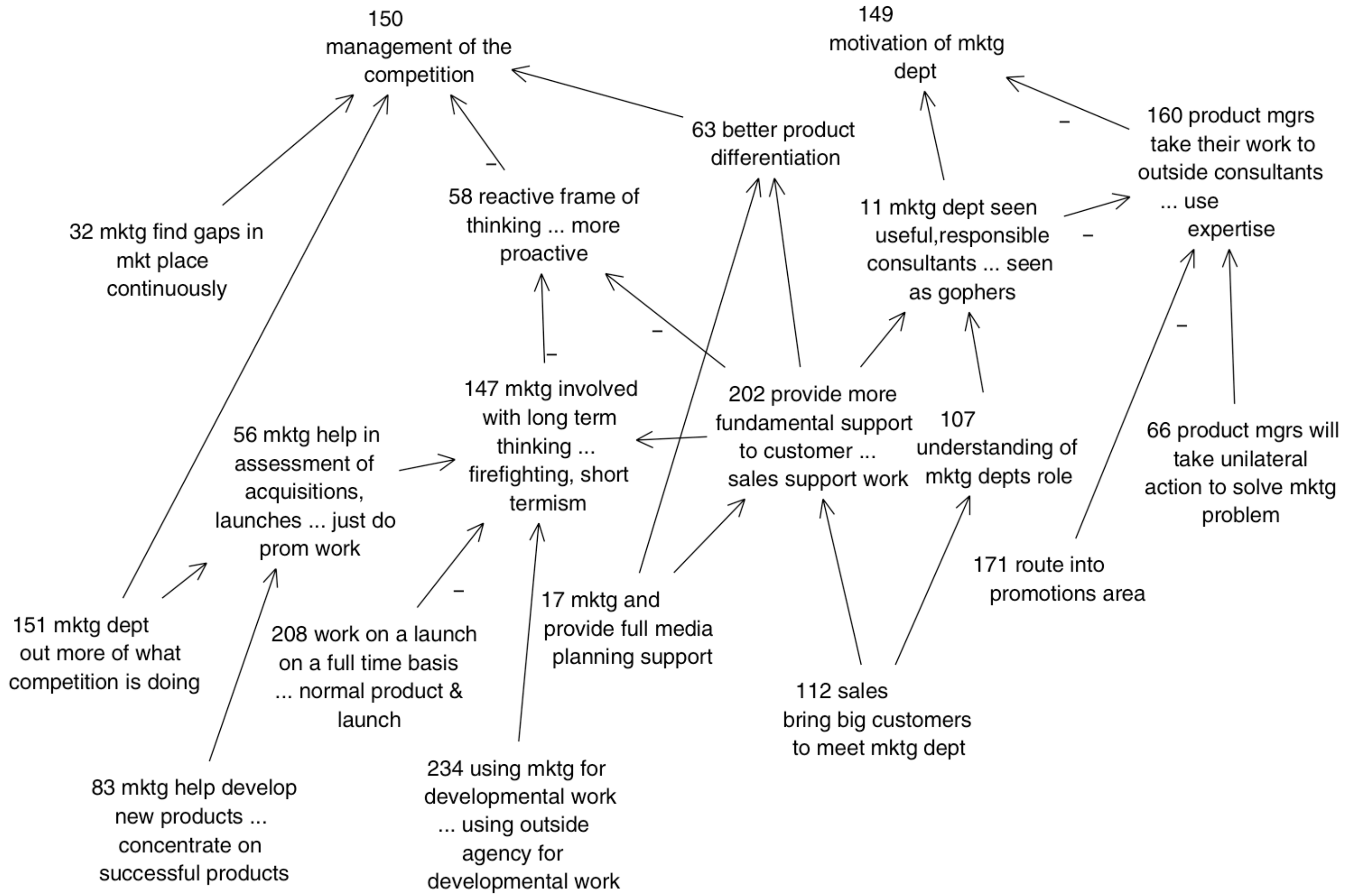
- understanding the information support needs of research scientists and technologists at a multi-national laboratory and manufacturer, and how its IT and information department should be re-organised (Checkland & Holwell, 1998);
- supporting the development of an automated and integrated wind tunnel system (Vidgen, 1997);
- representing the complexity and uncertainty of construction projects (Sutrisna & Barrett, 2007).

The rich picture approach is interesting to modellers of SoS situations as it appears to be versatile way of building up an understanding of a socio-technical situation. Whilst the approach is flexible and enables the representation of group processes and politics it does have limitations that may make it unsuitable for identifying threats and troubleshooting SoS situations. The first limitation is that it has a limited scalability as it does not support computer supported reasoning as neither the entities nor the relationships are formally defined. The second limitation is that the technique does not come with a set of principles to interpret how the different entities will influence each other and therefore it is up to individual interpretation and experiences to identify threats or causes of problematic behaviour.

These limitations have been recognised for some time by the IS community and attempts have been made to create tool support (Avison, Andrews, & Shah, 1992; J. Zhang, Smith, & Watson, 1997). Checkland has rejected this need for tool support on the basis that rich pictures are sketches for enabling sensemaking and communication rather than complete descriptions for the purposes of formal analysis.

2.4.5.2 Cognitive Maps

A cognitive map is a representation of a person's thinking about a problem that uses a directed graph to represent the situation (Eden, 1988; Eden, Ackermann, & Cropper, 1992). The nodes represent either a goal or a 'bipolar' description of a part of a situation that is believed to cause (or influence) another part of the situation. The directed links represent the believed direction of causality between nodes. There are two kinds of link between nodes. Positive links represent a causal relationship between the first pole of a node to the first pole of the node it is connecting to. Negative links represent a causal relationship between the second pole of a node and the first pole of the node it is connecting to - hence providing the notion of an inverse/negative relationship. See Figure 2.8 for an example of a cognitive map of a typical managerial situation.



150 management of the competition
149 motivation of mktg dept
63 better product differentiation
160 product mgrs take their work to outside consultants ... use expertise
58 reactive frame of thinking ... more proactive
32 mktg find gaps in mkt place continuously
11 mktg dept seen useful,responsible consultants ... seen as gophers
147 mktg involved with long term thinking ... firefighting, short termism
202 provide more fundamental support to customer ... sales support work
107 understanding of mktg depts role
56 mktg help in assessment of acquisitions, launches ... just do prom work
66 product mgrs will take unilateral action to solve mktg problem
171 route into promotions area
151 mktg dept out more of what competition is doing
208 work on a launch on a full time basis ... normal product & launch
17 mktg and provide full media planning support
112 sales bring big customers to meet mktg dept
83 mktg help develop new products ... concentrate on successful products
234 using mktg for developmental work ... using outside agency for developmental work
–
–
–
–
–
–
–

Figure 2.8 - A section of cognitive map of a managerial situation (Eden, 2004)

An essential feature of cognitive maps, that makes them distinct from influence diagrams, is their use of ‘bipolar’ statement nodes (Eden, 2004). Each statement node is composed of a ‘bipolar’ opposite that acts to highlight the potential variability in the situation e.g. ‘management involved in the long term thinking … fire fighting and short-termism’. The ‘…’ should be read as ‘rather than’ thus expressing a contrasting pole. This is intended to bring to attention the potential range of variables so that the situation can be modified to promote goal satisfaction.

Cognitive maps have been used extensively in action research projects to represent complicated / messy socio-technical situations and to make policy decisions. For instance they have been used to:

- identify and explore policy options within the Prison Service of England and Wales (Eden & Ackermann, 2004)
- understand the relationships between patient quality and hospital activities and programmes in the NHS (Telford, Cropper, & Ackermann, 1992)
- perform risk analysis of portfolios of complex projects (Ackermann, Eden, Williams, & Howick, 2007)
- perform stakeholder and conflict assessment in the context of natural resources management situations (Hjortso, Christensen, & Tarp, 2005), and complex projects (T. Williams, Ackermann, & Eden, 2003).

The cognitive map approach is potentially applicable by modellers of SoS situations. This is because the abstraction is both scalable and practical. The approach scales as computer aided analysis may be performed on the directed graphs to simplify models, highlight important clusters of interacting parts, virtuous or vicious circles, or other parts that metrics suggest may be of significance to an analyst. The approach is also practical as it can be used to understand the dynamics of complex situations and identify threats as illustrated by the above case studies.

The limitation of the cognitive map abstraction for modelling SoS situations is that it is purely a behavioural model. It has no means of representing a system’s parts and therefore it does not support the systematic elicitation of interactions based on a situation’s structure, context and processes. This is a rectifiable weakness as structural models can be produced using other approaches, which can be used to prompt the systematic identification interactions causes and effects. These interactions could then be represented and analysed using the cognitive map. For instance, it is plausible that a simple structural model such as a responsibility model could complement the systematic elicitation of cognitive maps.

2.5 Comparative Analysis

A diverse range of abstractions for representing systems were reviewed as part of this literature review. No single form of abstraction was found to be an off-the-shelf SoS modelling abstraction for socio-technical threat identification and troubleshooting. Therefore this section characterises the strengths and limitations of each of the approaches.

Table 2.1 - Summary of Strengths and Limitations of Abstractions for SoS Modelling

Approach	Functional Models	Agent Goal Models	Agent Responsibility Models	Cultural-Historical Activity Theory	Actor-network Theory	Rich Pictures	Cognitive Maps
Paradigm	Typically Machine Metaphor	Typically Machine Metaphor	Typically Machine Metaphor	Socio-Technical Metaphor	Socio-Technical Metaphor	Socio-Technical Metaphor	Dependent on Analyst.
Capable of describing the 'socio-technical' structure of a SoS	Partial	Partial	Partial	Yes	Yes	Yes	No
Capable of aiding in the identification of risks due to distribution across organisational / social / cultural / legal boundaries	Possible	Yes	Possible	Yes	Yes	Possible	Possible
Capable of aiding in the identification of human / organisational agents' potential to conflict / resist a system	Possible	Possible	Possible	Yes	Yes	Possible	Possible
Capable of describing 'large systems'	Yes	No	Possible	Possible	Possible	No	Yes
Capable of being analysable either algorithmically or quantitatively	Yes	Yes	Yes	No	No	No	Yes

2.5.1 A model capable of describing the 'socio-technical' structure of a system

CHAT, ANT and rich pictures all provide the capability to describe the socio-technical structure of a system. This capability is delivered by the fact that their descriptions of systems are textual, or textual and pictorial and thus may describe anything of interest to an analyst.

Functional models, agent goal models and responsibility models all partially provide the capability to describe the socio-technical structure of a system. These models are limited, or partial, as the kinds of entity that they are able to represent are limited by their choices of concepts for abstraction. Functional decomposition, agent goal models and responsibility models are limited as their capability to describe politics and groups processes is unclear and requires further research.

Cognitive maps do not provide the capability to describe a system's socio-technical structure. They describe the structure of behavioural interactions. E.g. how the performance of one activity influences another. This is in contrast to a structural description of a system as it would describe the logical relationships/dependencies between the parts that are producing the behaviour.

2.5.2 A model capable of aiding in the identification of threats that may arise due to a system being distributed across organisational / social / cultural / legal boundaries

Agent goal models, CHAT, ANT accounts are capable of aiding in the identification of risks arising from interacting technical and non-technical parts and there is research demonstrating their capability to identify risks specifically associated with crossing organisational/social/legal boundaries.

Agent goal models have also been used to analyse the influence of power, trust and social cohesion on systems that cross-organisational boundaries – see (Sutcliffe, 2011). This form of analysis appears limited for our purposes. Differences of organisational / social / cultural / legal norms were not explicitly considered although they appear important to analysing the success/failure of interactions across organisational boundaries.

CHAT has been used to understand the behavioural implications of parts being distributed across organisational / social / cultural / legal boundaries (Engestrom, 2001). CHAT was used to analyse incompatibilities, or tensions, between the objects and mediators of the interacting subjects and these identified tensions can be regarded as risks to the system.

ANT has been used to understand the behavioural implications of parts being distributed across organisational / social / cultural / legal boundaries (Mähring et al., 2004). ANT was used to analyse the failure of Denver International airport in terms of networks of interacting actors supporting the escalation of the project and then its implosion when additional actors destabilised the configuration.

Functional models, responsibility models, rich pictures and cognitive maps are all capable of aiding in the identification risks arising from interacting technical and non-technical parts. For these approaches however there is no research specifically demonstrating their capability to identify risks specifically associated with crossing organisational/social/legal boundaries.

2.5.3 A model capable of aiding in the identification of human / organisational agents' potential to conflict / resist a system

ANT and CHAT descriptions are capable of aiding in the identification of human / organisational agents' potential to conflict / resist a system and case studies demonstrate how this may be done.

Functional models, agent goal models, responsibility models, rich pictures and cognitive maps may be capable of aiding in the identification of human / organisational agents' potential to conflict / resist a system however this survey of the literature did not find case studies demonstrating how this may be done. This type of description and analysis may require a supplementary framework that indicates what kinds of social entity should be analysed to identify conflict / resistance.

2.5.4 A model capable of describing large systems

Functional models and cognitive maps are capable of describing large systems and case studies suggest they have been used to do so. The IDEF0 language is used by the US DoD agencies and is supported by major vendors such as IBM. Tool support is available that scales up to enable the visualisation and representation of large-scale systems engineering projects. Cognitive mapping software, such as Baxia Decision Explorer, has been developed to enable it use in commercial settings.

Responsibility models, CHAT descriptions and ANT descriptions may be capable of describing large systems. However this literature review did not find case studies suggesting they have the capability to describe systems with 1000s of socio-technical parts in a useful manner.

Agent goal models and rich pictures may not be capable of representing large systems. Agent goal models have been acknowledged to be limited in their ability to scale (Pastor, Estrada, & Martinez, 2011). Rich pictures are not hierarchical so the size of system that may be described is limited by the size of the drawing media. No evidence suggests that it would be possible to describe systems with 1000s of socio-technical parts in a useful manner using agent goal models or rich pictures.

2.5.5 A model capable of being analysable either algorithmically or quantitatively so that the maximum size and complicatedness of an analysable system is not restricted by human beings limited faculties

Functional models, agent goal models, responsibility models and cognitive maps are capable of being analysable either algorithmically or quantitatively and case studies demonstrate its use.

Functional models may be more formally represented, for instance as petri nets or as system dynamics models, and analysed to identify problematic interactions and run simulations – see for instance (Peters & Peters, 1997; Plaia & Carrie, 1995; Ruinan, Qing, Xin, & Qing, 2004).

Agent-goal models support computer analytics that enables the analysis of events on goal satisfaction (Asnar, 2009; Horkoff & Yu, 2009). These approaches are potentially highly scalable as goal models may be expressed in GRL (Goal Requirements Language) and algorithms applied to assess the satisfiability of goals in specific situations (Amyot et al., 2010). At present these algorithms only have partial support for cyclic graphs and thus do not have full support for non-linear interactions.

Responsibility models support basic computer aided analysis to help identify agents that may be overloaded or responsibilities that are unassigned (Lock et al., 2009). At present computer aided analysis does not support more sophisticated analysis such as identifying non-linear interactions.

Cognitive maps are perhaps the most promising type of model as they comprise directed graphs and thus are analysable using computational methods and support the analysis of non-linear behaviour See (Eden, 2004) for details on cognitive maps. See (Boccaletti, Latora, Moreno, Chavez, & Hwang, 2006) for a review of complex network analysis.

CHAT descriptions, ANT descriptions and rich pictures may not be capable of being analysable either algorithmically or quantitatively. Their style of representing a system in natural language or a picture does not lend itself to benefit from computer analytics (using current technologies) unlike more formal models. Analysis is therefore limited by the number elements a human analyst can comprehend.

2.5.6 Lesson Learned

Comparison of the abstractions' ability to scale and their available analytic support (in Table 2.1) suggests functional models and cognitive maps (directed graphs) may provide promising avenues for further research. The key limitation of the functional models (IDEF0 / FRAM) is that they would require to be *transformed* into petri-nets or system dynamic models for analysis. The key limitation of the cognitive map approach is that it does not provide a structural view of a system and so it requires a structural model of a system to be *transformed* into a cognitive map of the system.

Functional decomposition and agent responsibility decomposition may complement the cognitive map approach as they provide a sound basis for structural modelling and their models may be used to systematically elicit interactions between parts. For the purposes of SoS risk identification and troubleshooting, functional models, agent goal models and agent responsibility models are all viable candidates for being used as structural models. Functional modelling is a strong candidate since it has good tool support for visualising and managing models. The agent responsibility abstraction is also strong candidate due to its minimalist notation and its focus on analysing norms, which may be important to both analysing work across organisational boundaries and for identifying potential conflict.

2.6 Chapter Summary

In this chapter we reviewed existing approaches to representing and analysing socio-technical situations. We identified that within systems engineering there are two promising forms of decomposition namely functional decomposition and agent based decomposition. We identified the strengths and drawbacks of these approaches and recognised that there were a number of research opportunities with respect to representing and analysing politics and group processes within these modelling representations. It was also noted that further research with respect to analytical techniques that would support the analysis of large models would also be beneficial.

Outside of the field of systems engineering we identified that cultural historical activity theory, actor network theory, rich pictures and cognitive maps may provide interesting modelling frameworks that could be used to inform the modelling of SoS. Cultural historical activity theory is a promising and potentially scalable framework for analysing tensions that can exist within organisations and across organisations. Actor network theory provides a radical framework for analysing socio-technical systems that provides explanations that avoid invoking ‘social forces’ or other invisible social structures. Rich pictures provide a simple yet powerful approach of sketching out how social and political factors are influencing and problematising a situation. And finally cognitive maps illustrate the practical usefulness and analytic power of representing situations as directed graphs.

In the final section of this chapter we identified the strengths and weaknesses of the abstraction approaches and identified that cognitive maps (a directed graph approach) would provide a sound basis for a SoS modelling approach when complemented with an appropriate structural modelling approach that enables social analysis.

3. Research Methods

3.1 Introduction

"*If a thing can be observed in any way at all, it lends itself to some type of measurement method. No matter how 'fuzzy' the measurement is, it's still a measurement if it tells you more than you knew before*." (Hubbard, 2010, p. 3)

Measurement is used in a wide range of situations from understanding the usability of a device in a quantitative manner through to understanding why people do not use a system for the purpose it was intended. The purpose of this chapter is to review research methods that enable the study, or measurement, of socio-technical systems (including SoS) in situ. This means we will review methods or approaches that are available to researchers to collect and analyse data that enables the troubleshooting, or threat identification, of socio-technical systems that operate outside of controlled laboratory settings.

3.2 Data Collection

Data collection is the process by which a researcher attempts to collect evidence that reduces uncertainty over the answer to a research question. For example, when troubleshooting a system a researcher seeks gather evidence that reduces the uncertainty of whether a possible cause is a probable cause (Hubbard, 2010). This is usually achieved by gathering evidence that either excludes possible causes or that suggests probable causes.

Data collection methods must be carefully selected because they have limitations that may introduce bias or uncertainty into the data they collect. Investigations must be carefully designed to minimise the effect bias and uncertainty may have on the outcome of research (Creswell, 2009). The following section reviews commonly used approaches to data collection and identifies their strengths and limitations to indicate their suitability for the study of socio-technical SoS.

3.2.1 Surveys

A survey is "a well defined and well-written set of questions to which an individual is asked to respond" (Lazar, Feng, & Hochheiser, 2010, p. 100). Surveys are commonly used for the study of phenomena that involves large groups of people. For instance surveys are used to gather data to describe the behaviour of a population of employees with respect to their adoption of a system and its perceived success – see for instance (Delone & McLean, 2003; Venkatesh, Morris, Davis, & Davis, 2003). The strengths of the survey method are that for a low cost it can reach a large number of geographically dispersed people in a short period of time. In SoS contexts, this enables a single researcher to capture the ‘big picture’ in a short period of time.

Whilst surveys do scale up to large situations they suffer from a number of drawbacks. The first drawback of surveys is that they are typically self-administered so it is not possible ask immediate follow up questions or change the survey instrument once it has been sent. The second drawback is that they are not suitable for collecting ‘rich data’ as open questions often receive terse responses and surveys with many open questions are costly to analyse due to the time and resources necessary to read a large number of textual responses. The third drawback is that the validity of survey data is dependent on sampling variance and non-sampling errors such as response bias, non-response bias or recall bias (Assael & Keon, 1982). This can result in the introduction of uncertainty with respect to how representative the data is of the phenomenon being studied. Techniques have been developed to estimate the effects of these errors and therefore the level of uncertainty associated with a result from a survey can typically be bounded (Assael & Keon, 1982).

3.2.2 Interviews

Interviews comprise conversations and interactions with a participant with the aim of receiving responses to a set of questions that are intended to generate data about a phenomenon being studied. Interviews are a commonly used research method for the study of phenomena that can be understood by means of collecting ‘in-depth’ data from a small number of individuals (Lazar et al., 2010). For instance they can be used to understand a person’s view of a problem, its context and explore a wide range of concerns with unforeseen follow up questions. This enables the interviewer to capture data that would be difficult to capture using less costly approaches such as questionnaires.

Whilst interviews enable the capture of data that would be difficult to capture using surveys, they suffer from a number of notable drawbacks. Firstly, they are costly to perform as an interviewer must meet each interviewee and spend a period of time asking and answering questions. This means that the method is costly to scale-up to situations where a large number of participants are involved. This limitation can be addressed to some extent through the use of focus groups. However, this approach has its own drawbacks since it may introduce response bias due to people responding in the presence of others.

The second drawback of interviews is that the analysis of the data they produce is costly since transcribing recordings or writing up interview notes is time consuming. A third drawback, similar to that of surveys, is that the validity of data is dependent on sampling variance and non-sampling errors. In some situations, this limitation can be minimised by combining interviews with observations so as to detect recall bias if it occurs. Of course, the observations themselves are subject to response bias since the individual may act differently whilst being observed –

this is known as the observer-expectancy effect. Observer-expectancy effects can be identified using additional techniques such as concealed observation to see if the subject's behaviour changes when they believe they are not being observed.

3.2.3 Ethnography and its scalability

Ethnography is a research method used by anthropologists and sociologists to study people in their naturally occurring settings through the use of observations, interviews and participation (Lazar et al., 2010; Rouncefield, 2011). Ethnography is founded on the notion that to understand complex human activities in-depth *participative* study is required. Participative study is intended to overcome the limitations of surveys, interviews and observational studies. This is because participation minimises subject recall bias as the participant observer may observe the subject performing tasks and may also learn how to perform the task for them self thus triangulating their findings from two sources. Participative methods may also reduce response bias because the researcher spends a long time onsite and so the researcher is perceived to be part of the group, hence observer-expectancy effects are minimised.

The main limitation of ethnography for studying SoS is that it is a time consuming research method (John Hughes, King, Rodden, & Andersen, 1995). An ethnographer must visit each geographical location where a socio-technical component is present, they must attempt to embed themselves within that context, and participate in the activities of that community or group. This may require a large team of ethnographers to perform, especially if a system is geographically dispersed. Once this has been performed the textual descriptions of each group must then be analysed and aggregated. This again is a time consuming process, especially if multiple researchers perform the ethnographies.

Attempts have been made to lower the cost of ethnography via the use of "quick and dirty ethnography" or "rapid ethnography" (John Hughes et al., 1995; Millen, 2000). The quick and dirty approach comprises short focused studies to quickly gain the general picture of a setting. The phrase "quick and dirty" is used because the duration of the fieldwork is of a short (weeks or months) duration relative to the phenomenon of study and that at the outset it is recognised that gathering a complete and detailed account of the phenomenon is highly unlikely. To use the duration of fieldwork as efficiently as possible, 'quick and dirty' ethnographies are theoretically informed by drawing on insights from previous studies (John Hughes et al., 1995). The primary purpose of the 'quick and dirty' approach is to focus on portions of the phenomenon that are likely to inform strategic decision-making rather than being exploratory as per traditional ethnography.

Rapid ethnography is similar in nature to quick and dirty ethnography. The approach suggests that the fieldwork should have a narrow focus, it should use multiple interactive observational techniques and should use collaborative and computerised data analysis methods (Millen, 2000). To speed up analysis it is suggested that field site experts guide the analyst or act as a "field guide" (Millen, 2000), that “fringe members” should be focused on as they move freely around groups and that multiple analysts can be used in the same field site.

Rapid or quick approaches also suffer limitations once the scale of a situation is increased. As the scale increases the depth to which participants may be studied in a given time and the representativeness of selected participants become issues that limit the validity of the study (John Hughes et al., 1995). These issues are not problematic when studying single sites such as control rooms but become troublesome when studying multiple interacting sites as the researchers’ decisions on where to be, when and for how long, may result in accounts that are not representative of the phenomenon of study.

Although ethnography is more time intensive than the use of surveys and interviews, surveys and interviews are more limited in their ability to elicit ‘in-depth’ data. This can make them unsuitable data collection methods for troubleshooting certain situations. This implies that there is a time cost vs. completeness trade-off when deciding between ethnography and other methods. This therefore poses an interesting and extremely challenging research question:

- ‘what is the most appropriate data collection and analysis *strategy* for troubleshooting or identifying risks in a SoS situation?’

I am *prima facie* inclined to hypothesise that one should adopt the strategy of using cheaper methods to develop a ‘big picture’ view and use ethnography sparingly in areas that the big picture data suggests finer grain data may be required. This, of course, would result in theoretically informed, multi-site, multi-method studies that may include the strategic use of ethnography.

Other researchers when faced with research questions of a similar scale have suggested similar research strategies. For instance (Pollock & Williams, 2010, p. 531) have proposed a need for “strategic ethnography - whereby the choice of research settings and the scope of the studies is informed by provisional theoretical/empirical understandings of the locales … as well as by the specific research concerns and issues under examination”. Their proposal is motivated by the fact the studies of large-scale systems in science and technology studies have typically been single site studies over a short duration and thus have given piecemeal accounts of what they see to be a multifaceted historically evolving phenomenon. They also acknowledge that performing these studies will be non-trivial as it will require team projects or even community sized efforts to resource these studies.

Similarly (Hine, 2007, p. 618) has proposed the notion of connective ethnography to facilitate the study of e-science’s “diverse sites, connected in complex and heterogeneous ways.” Connective ethnography comprises multi-site ethnography that endeavours to explore global connections suffusing sites. This is achieved by visiting and participating in both online and offline sites to understand the cultural activities that enable infrastructure assist in the users meaningful activities as e-scientists. Other approaches such as network ethnography (Howard, 2002) have also been proposed.

Network ethnography (Howard, 2002) is the process of using ethnographic field methods on cases and field sites selected using social network analysis. It is a particularly interesting approach from a SoS perspective as the gathering and analysis of social network data can be particularly low cost and its enables the management of sample bias as might appear when selecting informants using snowball sampling or similar approaches. Additionally, the researcher can use the data to identify communities and well connected individuals that may be able to arrange meetings with other important figures. Another attractive characteristic is that the researcher can monitor the dynamics of the situation over time by monitoring the changes in communities, important figures and interactions.

3.3 Data Analysis

Data analysis is the process by which a researcher interprets the data they have collected to determine whether it reduces uncertainty over the answer to their research question. Data can be analysed in a quantitative manner and in a qualitative manner. Analysing data in a quantitative manner means that the data is interpreted and represented numerically and is analysed using statistical techniques. The outputs of these statistical techniques suggest what can be inferred from the gathered data with a certain level of confidence. An alternative to quantitative analysis is qualitative analysis where data is treated as units of linguistic meaning, and analyses are performed in an attempt to identify patterns, interconnections or themes. The outputs of qualitative analysis may be used to suggest what kinds of description of a situation are more plausible than others. In this section, we review a number of quantitative and qualitative approaches that may be useful for the analysis of SoS situations.

3.3.1 Quantitative Approaches

Multivariate statistical analysis is a useful approach for analysing and understanding large-scale phenomena where quantitative data may be gathered. Multivariate analysis refers to statistical techniques that simultaneously analyse multiple measurements of the phenomenon of interest (Hair, Black, Babin, & Anderson, 2008). Multivariate techniques are often extensions of bivariate techniques – for instance a regression with one predictor variable is extended in the multivariate case to include multiple predictor variables. Multivariate techniques may be split into two groups based on their purpose:

1) interdependence techniques - that aim to analyse the extent that a set of variables are interrelated.
2) dependence techniques - that aim to analyse the extent that a set of independent variables can predict/explain a set of dependent variables.

Factor analysis (including principal component analysis) is an interdependence technique that can analyse the interrelationships between large numbers of variables and explain these variables in terms of their common underlying dimensions (or factors). The purpose of this technique is to understand the factors that influence a broad set of variables. For instance, it could be used to identify 'critical success factors' in a portfolio of successful projects.

Dependence techniques analyse changes to a dependent metric variable in response to changes to independent metric variables. These techniques are useful for the analysis of SoS as they enable a researcher to predict the effect of changing an aspect of a situation on variables of interest. In situations where the single dependent variable is dichotomous (Yes/No) or multichotomus (low-medium-high) techniques such as multiple discriminant analysis may be used (Hair et al., 2008). This enables the prediction of the likelihood that an entity will belong to a particular category based on several independent variables.

Logistic regression may be viewed as an extension of this technique as it enables a single dichotomous or multichotomus dependent variable to be predicted by any types of independent variables regardless of whether they are metrics, dichotomous or multichotomus. Canonical correlation may be viewed as a further extension of multiple regressions enabling the correlation of multiple dependent metric variables based on multiple independent metric variables.

Structural equation modelling (SEM) and confirmatory factor analysis (CFA) are further techniques that are widely used as they allow multiple dependent variables to be predicted by multiple independent variables (Hair et al., 2008). SEM may be viewed as providing a technique for simultaneously estimating a series of separate multiple regression equations. A SEM analysis comprises two components: the structural model that describes the paths between dependent and independent variables; the measurement model that enables the use of several indicators of a single independent or dependent variable. CFA enables the researcher to assess the contribution of each indicator to a variable as well as determine how reliably the indicator measures the concept represented by the independent / dependent variable. In a SoS situation this technique might be used to develop and validate a model that predicts whether a person or team is likely to resist, support or be neutral towards a particular system based on their responses to a survey.

Network analysis is also an interesting quantitative approach for analysing and understanding large-scale situations. Network analysis enables the analysis of systems with multiple interconnecting parts if their interconnections are known. Network analysis has enabled the analysis of systems of thousands to millions of nodes such as electric power grids, the Internet and social networks. This ability to analyse vast numbers of nodes and interactions makes network analysis attractive for understanding large-scale situations.

There exist a number of candidate metrics that may lend themselves to understanding SoS situations. Firstly, the 'influence' or 'importance' of an element (node) in a socio-technical system (graph) may be identifiable using network centrality metrics. Network centrality metrics have been used for this purpose in other domains such a social network analysis (Boccaletti, Latora, Moreno, Chavez, & Hwang, 2006). Secondly the 'complexity' (interconnectedness and extent of feedback) of an element may be estimable by combining existing techniques to count the number of loops a node is involved in and its number of links to other nodes. The counting of loops, referred to as k-cycles, is a well-established practice (Vázquez, Oliveira, & Barabási1, 2005) as is counting the links of node (Opsahl, Agneessens, & Skvoretz, 2010).

In anthropology, network analysis has also been used to aid the analysis of ethnographic data. Ethnographic materials can be coded into network databases so that they can be analysed using techniques from network analysis (White & Johansen, 2005). The use of network analysis in this manner has enabled researchers to understand how local interactions (observed using ethnography) form networks of interactions that have a macro structure with global properties that alter the context of interactions thus providing an understanding between the dynamics between micro-level interactions and macro level phenomena. This use of network analysis is again a particularly interesting approach from a SoS perspective as understanding how micro-level interactions induce macro level phenomena is important to both troubleshooting systems and identifying threats to systems.

3.3.2 Qualitative Approaches

Qualitative data analysis is often used in situations where a researcher wants to understand a system's social and cultural context (Myers & Young, 1997). This is useful when a researcher wants to:

- understand how a system's stakeholders perceive and evaluate a system;
- understand the influence of social and organisational context on system use;
- explore causal mechanisms for the phenomenon of study;
- provide formative evaluation that is aimed to improve a system underdevelopment;
- contextualise quantitative evaluation (Kaplan & Maxwell, 2005)

Qualitative analysis may be characterised by its reliance on a researcher's interpretation of a text, or text analogue such as an interview transcription or observational data, to obtain a sense of a phenomena being studied (Creswell, 2009). Qualitative analysis is often associated with:

- the interpretation of data collected from humans in their natural settings e.g. their places of work, or in the geographical location of the phenomena of interest;
- data collection prior to hypothesis formation;
- evolving research designs and sampling strategies;
- studying subjects' perceptions of a phenomena by understanding and negotiating meanings and interpretations;

- utilising tacit knowledge to make sense of the nuances and differences between subjects' multiple realities;
- using coding to develop themes and identify interactions between themes to facilitate a researcher's sense-making of the phenomena.

Although making clear distinctions between different modes of qualitative data analysis is difficult, one may characterise them according to three broad types: (1) grounded theory; (2) hermeneutics; (3) narratives, metaphor and semiotics (Myers, 1997; Myers & Avison, 2002).

3.2.1 Grounded Theory

Grounded theory (Corbin & Strauss, 2008) is an inductive method of research that attempts to generate theories that are grounded in the qualitative data gathered during a study. Unlike experimental research that begins with a hypothesis to test, grounded theory begins with the collection of data and uses this data to influence the ongoing design of the research and also the research outcome e.g. a theory to explain an emerging research question (Jim Hughes & Jones, 2003; Matavire & Brown, 2008).

Grounded theory approaches analyse data using the notions of *constant comparison*, *coding* and *theoretical sampling* (Corbin & Strauss, 2008; Jim Hughes & Jones, 2003; Matavire & Brown, 2008).

Constant comparison means using comparison to evaluate data sources against each other data in order to determine their accuracy and generalisability, in addition to identifying concepts and verifying theory.

Constant comparison is performed using: (1) *open coding*, which facilitates development of concepts and categories; (2) *axial coding*, which identifies relationships between categories; and (3) *selective coding*, which means to cease open coding and restrict coding to categories that are relevant to the formation of theory. The categories that remain should explain variation and account for contradicting evidence[5].

Coding is a technique for deriving concepts that emerge from data and developing those concepts so as to understand their properties (Lazar et al., 2010). Concepts emerge by viewing/reading the data and making comparisons between data so that commonalities and distinctions become apparent and concepts emerge.

Depending on the style of grounded theory adopted, the data-coding scheme may be constructed on an ongoing basis, or may be an '*a priori* coding scheme' based on an established theory or framework. Once coding is complete another party may check the reliability of the coding. The coding process is often supported by software packages that enable the storage of data in a searchable format, thus

[5] It should be noted by the reader that grounded theory is not a unified approach and that there exist at least two differing accounts of the approach – commonly referred to as Glaserian and Straussian camps. For discussion on the differences between the camps refer to (Matavire & Brown, 2008).

enabling a researcher to code and compare data in a manner that can be more efficient than hand-coding. Popular software packages include Atlas.ti[6] and QSR NVivo[7].

Theoretical sampling is the notion that a grounded theory sampling strategy is influenced by an emerging research question and theory, rather than using a pre-determined sample strategy as per traditional quantitative research. This means that a researcher may alter their sampling method to ensure that they are able to validate or falsify their theory using saturation sample.

The grounded theory method of qualitative research is considered to be particularly well suited for the analysis of systems (Baskerville & Pries-Heje, 1999; Jim Hughes & Jones, 2003). This is because it encourages the creation of theory rather than detailed description. This is considered advantageous since theory may be used to guide practice e.g. troubleshooting a situation (Baskerville & Pries-Heje, 1999; Kaplan & Maxwell, 2005). Grounded theory also provides a rigorous way of analysing data for the purposes of action research (Baskerville & Pries-Heje, 1999; Wastell, 2001). Grounded theory has been used to study many systems and contexts including the study of system adoption (Orlikowski, 1993), the success and failure of knowledge management systems (Wastell, 2001), the evaluation of IT systems in the public sector (Jim Hughes & Jones, 2003).

The main disadvantage of the grounded theory approach is that it relies on the expertise of the researcher not to inject bias, and their open-mindedness to produce reliable and useful findings that may be used to inform a situation. Also it should be noted that grounded theory can be at odds with interventionist systems research as grounded theory tends to be broad exploratory technique. However in practice, studies suggest that this theoretical limitation is not problematic as researchers adapt the scope and nature of the grounded theory approach to suit their own purposes (Baskerville & Pries-Heje, 1999; Matavire & Brown, 2008).

3.2.2 Hermeneutics

Hermeneutics is a mode of analysis that is concerned with the *meaning* of text or a text analogue (Myers, 1997, 2004). Hermeneutic analysis attempts to make sense of a text, or text analogue, that is confused, incomplete, messy, or seemingly contradictory. Although there are a variety of hermeneutic styles from differing philosophical traditions, broadly speaking hermeneutic analysis attempts to elicit the underlying meaning and coherence of a text using five key principles: (1) *historicity*; (2) *hermeneutic circle*; (3) *prejudice*; (4) *autonomization* and *distanciation*; (5) *appropriation* and *engagement* (Myers, 2004).

Historicity is the notion that the way a person interprets and acts is influenced by their past experiences, and that their past experiences are shaped by the habits and

[6] For further details on the capabilities of Atlas.ti see http://www.atlasti.com/

[7] For further details on the capabilities of QSR NVivo see http://www.qsrinternational.com/products_nvivo.aspx

practices of the communities they have interacted with (Myers, 2004). This implies that a person's interpretation of the present day, such as a problematic information system, is historically informed and that to understand a person's interpretation of a situation requires an understanding their historicality.

The *hermeneutic circle* refers to the proposition that the meaning of a text is elicited by interpreting the text as a whole and as a set of parts – this iterative process is referred to as a hermeneutic circle (Myers, 1997, 2004). The hermeneutic circle is based upon the premise that there is dialectic between the understanding of a text as a whole and the interpretation of its parts. It is postulated that the anticipation of the meaning of the whole shapes the interpretation of the parts. Similarly the meaning of the parts shapes the interpretation of the whole. Therefore to understand a text an interpreter must iterate between studying the parts and studying the whole until a coherent meaning can be elicited.

Prejudice is the notion that prior knowledge plays an important role in an interpreter's understanding of a text (Myers, 2004). It is postulated that an interpreter's attempt to understand a text always involves some prior knowledge or expectation. E.g. a person requires prior knowledge of the language a text is written in, the domain specific terms, and its historical context. Prejudice also implies that an analyst requires an awareness of their own historicality, and that it is necessary to reflect on how it may be shaping their understanding of a text. In the context of studying information systems this means that when attempting to understand a problematic system a researcher must not only be aware of his subjects' historicality but also his own when attempting to understanding the problem.

Autonomisation and *distanciation* are notions that describe the properties of a written text (Myers, 2004). Autonomisation refers to that fact that once an author writes a text its becomes autonomous in the sense that the meaning of the text exists independently of the author and the author may no longer control the context in which it is interpreted. Distanciation refers to the distance in time, geography and social space between the original author and reader of the text. In the context of studying information systems this means that the meaning/purpose of a system changes over time and space. For instance, the outcome of an information system development becomes distanced from its original purposes or justification for development, and thus may be interpreted in multiple ways by multiple interpreters.

Appropriation and *engagement* is the notion that a reader can only understand the meaning of a text if the reader appropriates its meaning (Myers, 2004). That is to say that the readers must make the text their own by critically engaging with the text. In the context of information systems this means that a researcher attempting to understand a problematic system needs to critically engage in the problem by iteratively forming hypotheses, searching for falsifying evidence and reformulating hypotheses until a theory (or explanation) of the problem can be generated that accounts for apparent contradictions or tensions between different stakeholders' perspectives.

Hermeneutics has been used in a number of case studies to understand messy situations involving information systems. For instance the study of the social and political context of email communications by managers in a corporation (Lee, 1994), the study of power, assumptions and hidden agendas in a large public sector IT project (Myers & Young, 1997), the analysis of myths concerning call centre based customer services (Corea, 2006), the analysis and amelioration of information systems development practices (Boland, Newman, & Pentland, 2010; Butler & Fitzgerald, 1997).

The strength of hermeneutic approaches are that they provide the researcher with freedom to pursue anomalous findings and enables them to put their interpretation on the data rather than simply attempting to determine the meaning of a text (Cole & Avison, 2007). Another strength of hermeneutic research is that researcher 'bias' is documented and used in an explicit and systematic manner to generate insights. The researcher builds a framework of understanding that outlines their set of assumptions, concepts and practices that constitutes their way of viewing reality. The prejudices that structure the construction of meaning are explicitly documented when identifying themes for discussion during interview and also when providing ideas for coding.

The disadvantages of hermeneutic approaches are that, unlike grounded theory, studies do not aspire to build theory to explain the relationship between a set of propositions that have been repeatedly tested (Cole & Avison, 2007). Hermeneutic approaches are also difficult to learn as there is not a single well-defined approach, and their focus on accounting for anomalies may also be at odds with troubleshooting and interventionist style research.

3.2.3 Narrative, Metaphor and Semiotics

Narrative, metaphor and semiotics are modes of analysis that attempt to understand the nature of IS related social practices by understanding the nature of the stories, metaphors or symbols that are used by participants describing the practice (Myers, 1997; Myers & Avison, 2002). Stories, metaphors or symbols are studied because they facilitate the communication of meaning between individuals and are critical for organisational sense-making and therefore their study may impart insights (Hirschheim & Newman, 2002). The state and popularity of this research is difficult to ascertain as surveys and criticism of this approach of studying of information systems is non-existent or too superficial to enable a detailed discussion in this thesis.

3.4 Chapter Summary

Our review of research methods indicates that to study SoS, in a cost effective manner, the use of research strategies that manage a time cost vs. completeness trade-off is essential. In order to meet this time cost vs. completeness trade-off our research suggests the use of theoretically informed multi-site, multi-method studies. During the review we observed that low cost methods such as surveys and focus groups might be suitable for eliciting the ‘big picture’ context of a SoS and that this information could be used to focus further efforts. We also noted that qualitative data analysis might be a useful approach for analysing data as it enables analysis of the social and political context of SoS as perceived by different stakeholders.

4. A Modelling Abstraction for Socio-Technical Troubleshooting and Threat Analysis

4.1 Introduction

The aim of this chapter is to formulate a basic modelling abstraction for socio-technical troubleshooting and threat identification of SoS. To achieve this aim we:

(1) identify a practical strategy for analysing the behaviour of SoS that copes with under-specification and low comprehensibility;

(2) identify information requirements and practical requirements of the above strategy;

(3) develop a responsibility-based modelling abstraction that meets these requirements;

(4) compare the responsibility-based abstraction with other prominent modelling abstractions.

This chapter is structured so that in section 2 we seek to identify a practical strategy for analysing the behaviour of a SoS. We begin by considering the challenges of modelling SoS that are under-specified and that have a low comprehensibility. We then propose that a strategy of creating models that *bound* the possible behaviour of a SoS may be useful and realistic. We then postulate that high-level structural models may be sufficiently invariant and informative to enable analyses that bound the behaviours of a SoS.

In section 3, we propose six classes of information and two practical requirements to ensure that the modelling abstraction captures the necessary information to enable analysis of SoS. In section 4 we use an example SoS to illustrate that the responsibility modelling abstraction meets the requirements set out in section 3 and thus may be used for SoS socio-technical threat identification and troubleshooting. Our example comprises a simplified SoS, called 'MegaFileShare', that is implemented using a number of cloud-based systems. The case study in this chapter is intended to be illustrative, rather than thorough, as subsequent chapters will contain such analyses. The purpose of this example is to simply demonstrate that it is feasible that the notion of responsibility modelling (supported by causal models of risk clauses) meets these requirements.

In section 5, we explain that responsibility modelling is a promising abstraction for troubleshooting and threat identification because of its *stakeholder centric* focus. Rather it focuses on desirable and undesirable behaviours and their *contextualised* importance to each stakeholder in a model. The responsibility abstraction enables the analysis of which stakeholders are ultimately liable for what system behaviours, thus enabling stakeholders in boundary crossing systems to understand the risks they are accepting and those that they are transferring to other agents. Section 5 also explains that responsibility modelling is of practical use as responsibilities tend to be unproblematic to elicit and that the notions used in responsibility modelling are grounded on notions used in the behavioural sciences to predict human behaviour. This is notable as it enables the analysis of 'socio' issues using insights from the behavioural sciences.

Finally in section 6, the responsibility abstraction is contrasted with other abstractions including IDEF, UML and I* to demonstrate how their affordances compare, and to what extent they are able to meet the information and practical requirements defined in section 3.

4.2 A Strategy for the Socio-Technical Analysis of a SoS

SoS may be seen as a difficult class of system to model and analyse. This is because they are challenging to describe and understand due their *under-specification* and *low comprehensibility*. They are under-specified because:

- their specific operating conditions and exact configuration are too vast or ephemeral to completely specify (Hollnagel, 2008; Montibeller & Belton, 2006).
- their parts and interactions change over time and thus creating/maintaining complete structural and behavioural descriptions is too costly/difficult (Hollnagel, 2008).
- the number and nature of parts and interactions may be disputed, or no one person has a complete view of the system, and so multiple potentially inconsistent views of the system may exist.

SoS have a low comprehensibility because:

- interactions between system parts may be non-linear and emergent (Hollnagel, 2008)
- no individual stakeholder has a complete view of the system and its operating conditions
- stakeholders may need to make sense of a large number of parts, interactions and views to understand the system's structure and behaviour.

Whilst acknowledging these challenges, an analyst attempting to understand the socio-technical behavioural properties of a SoS must adopt a strategy of analysis that 'makes the best of a bad lot'. Since it has been acknowledged that creating and maintaining descriptively complete structural and behavioural models of a SoS is unrealistic we must instead set our sights on a more modest strategy.

One such strategy is to develop models that attempt to *bound* the possible behaviour of a SoS. Using this line of reasoning one can attempt to identify attributes of SoS that are sufficiently high level to inform the analysis of the systems behavioural properties (as a totality) and are sufficiently invariant so that once information is collected about these attributes it is not ‘out-of-date’ prior to analysis.

The structural attributes of SoS, such as the dependencies between high-level components, may be sufficiently high-level and invariant to enable analysis that bounds their behaviour. We know from the study of complex networks that the analysis of structural properties of network systems (including social networks) can enable the bounding of their behavioural properties in certain situations e.g. percolation, epidemic spreading, diffusion (Boccaletti et al., 2006; Newman, 2003; Vespignani, 2012). We also know from prior research on I* models, and agent responsibility models, that potential system behaviours (e.g. threats) may be identified by inspecting their structural properties (Asnar, 2009; Asnar & Giorgini, 2006; Lock et al., 2009; Sommerville et al., 2009).

To *systematically* identify the potential behaviour of SoS, I suggest that the development of a structural model is of practical importance. Whilst it may be possible to create behavioural models, such as cognitive maps or system dynamic models, without first creating a structural model, I propose that this is unwise as it relies upon stakeholders and subject matter experts having correct mental models of the system’s structure. I do not want to make this assumption as it is acknowledged that SoS are vast and that people may have differing mental models of these systems. Therefore by creating a structural model I want to ensure that inconsistencies between stakeholders’ high-level views of a system are identified, discussed and resolved (or are at least ‘in the open’) prior to analysis beginning.

To *systematically* identify potentially complicated linear and non-linear behaviours of a SoS, I suggest that the development of a behavioural model is also of importance. The purpose of the behavioural model is to understand how the identified potential behaviours may combine and perhaps create complicated linear or non-linear behaviours. I suggest that the creation of a high-level behavioural model is of particular importance since we have acknowledged that SoS have a low comprehensibility and therefore we cannot assume that a stakeholder, or even a subject matter expert, will have a mental model of all the possible behaviours and their interactions.

A crucial aspect of the bounding strategy with respect to its practicality is the extent that these structural and behavioural models will need to be complete descriptions of a SoS. I hypothesise that this will depend upon the purpose of the analysis and the nature of the system being analysed. It may be the case in some situations that a small amount of information about the structure of a system is sufficient to create a behavioural model that bounds its behavioural properties to within a range that meets an analysts needs e.g. a small amount of information about the structure of a system may enable the analyst to a claim that a system has a property p. The case studies in this thesis shed light on the extent descriptions may be high level and incomplete, and the kinds of claims that may be made.

4.3 Requirements of the Modelling Abstraction

To execute the strategy described above, a modelling approach is required to capture the necessary information and represent it in a manner that makes it amenable to analysis. Below I propose that the required modelling abstraction must meet the following information requirements and practical requirements to ensure its scalability.

1. It must capture the following classes of information:
    A. desirable and undesirable behaviour;
    B. the consequences of undesirable behaviour for a component and its interdependent components;
    C. the consequences of desirable behaviour not occurring for a component and its interdependent components;
    D. the person/organisation that is accountable for the (non)occurrence of behaviour;
    E. the resources that a component depends on to perform the behaviour that is accountable for;
    F. the dependencies between system-level behaviour and component level behaviour.
2. The model should have the potential to represent systems with non-trivial numbers of components and interactions.
3. The analysis of the models should have the capability to be algorithmic, or computer assisted, such that the number of components a person can interpret does not limit the scale of an analysable system.

Requirement 1-A is important because it enables undesirable and desirable behaviours to be compared to the system's potential behaviour. This comparison enables the identification of threats from undesirable behaviours that are potential system behaviours and desirable behaviours that are not potential behaviours.

Requirements 1-B and 1-C are important because the consequences of undesirable behaviour on behavioural properties of interest need to be understood so that threats are identified. It is also vital to capture the consequences of desirable behaviour not occurring so its impact on system behaviour may also be assessed.

Requirement 1-D is important because the capture of which human/organisation is accountable for certain behaviours, provides insights into the differing interests of the human/organisational agents composing the system. This is particularly important in assessing systems that cross organisational boundaries as 'who is responsible for what', and what the rewards/repercussions are if desired behaviours are unmet becomes important.

Requirement 1-E is important because agents may depend on certain resources to perform behaviour that they are accountable for. To troubleshoot and perform threat analysis an analyst requires information on these dependencies because their absence or variance may be a cause of undesirable behaviour.

Requirement 1-F is important for the purpose of troubleshooting and threat analysis because an analyst needs to trace component level behaviours to system level behaviours and vice versa. Troubleshooting can be thought of as tracing problematic system level behaviours (or properties) to component level behaviours (or properties). Threat analysis can be thought of as tracing component level behaviours (or properties) to system level behaviours (or properties).

Requirement 2 is important because an abstraction that does not enable the representation and analysis of a large number of components and interactions may not be practical for the analysis of SoS.

Requirement 3 is important because an abstraction that is limited by the number of nodes an unaided person may analyse and understand in a timely manner may not be practical for the analysis of SoS.

4.4 Responsibility Modelling

Responsibility modelling has been proposed by several researchers as a useful abstraction for analyzing the dependability of socio-technical systems (Blyth, Chudge, Dobson, & Strens, 1993; Dobson & Martin, 2007; Strens & Dobson, 1993). I use responsibilities as part of a graphical modelling notation that represents 'responsibilities', 'agents' and 'resources' interconnected by relationships.

For the purposes of responsibility modelling, a responsibility is defined as:

"A duty, held by some agent, to achieve, maintain or avoid some given state, subject to conformance with organizational, social and cultural norms" (Lock et al., 2009)

The term duty in this definition captures *obligation* and *accountability* aspects of responsibilities such that if an agent does not appropriately discharge their obligation they will be held *liable*. The phrase conformance with organisational, social and cultural *norms* captures the fact that responsibilities must be discharged in accordance with legal and domain standards.

For the purposes of modelling the SoS analysed in this thesis we use the following entities and relationships:

- Responsibility: An entity representing a duty to achieve, maintain or avoid a specified state subject to conformance with norms.
- Resource: An entity representing physical material or information that contributes to meeting an obligation e.g. documents, databases, servers, tape drives, machines.
- Human Agent: An entity representing a human being often referred to by their role e.g. Support Manager.

- Organisational Agent: An entity representing an organisation e.g. an enterprise or government agency.
- Responsibility For: A relationship representing the allocation of a responsibility to an agent
- Has: A relationship representing the allocation of a resource to an agent or responsibility
- Depends: A relationship representing that the fulfilment of a responsibility is dependent on the fulfilment of another
- Association: A relationship representing that an entity is related to another. The association relationship may be annotated to clarify the relationship if necessary.

To identify threats responsibility modelling is combined with a HAZOPS style approach to identifying risks (Sommerville et al., 2009). Threats are identified via the means of 'risk clauses' – see for an example see Table 4.4 in this section. Risk clauses are composed from a target, a hazard, a condition, a set of consequences/threats and an estimation of their severity and likelihood (when appropriate).

Definitions of these terms are provided below.

Target: The entity or relationship to which the risk clause refers. E.g. an entity may be the responsibility to support and maintain a virtual machine instance. A relationship may be the allocation of a responsibility or resource to an agent.

Hazard: Using a restricted set of keywords we aim to provide a checklist of hazard source categories to consider. The hazard keywords we used are outlined below:

- Early - Occurrence of entity/relationship before required.
- Late - Occurrence of entity/relationship after required.
- Never - Non-occurrence of entity/relationship.
- Incapable - Occurrence did not take place although attempts were made to fulfil the obligation.
- Insufficient - Occurrence of the entity/relationship at an incorrect level.
- Impaired - Occurrence of the entity/relationship in an incorrect manner.
- Changes - The entity/relationship changes on a 'permanent' basis.

Condition: A description of the potential conditions that could manifest as a result of the hazard category considered.

Consequences: A threat to a behavioural property of the system e.g. availability, reliability, safety, integrity, or maintainability.

Severity: Liabilities resulting from the hazard manifesting.

To illustrate that responsibility modelling provides a suitable abstraction for modelling SoS for the purposes of troubleshooting and threat identification. Responsibility modelling will be performed on a simple example of a SoS called 'MegaFileShare'. 'MegaFileShare' consists of a website for sharing files and serving advertisements to users as they consume the content.

The 'MegaFileShare' example demonstrates how desirable and undesirable behaviour and their consequences are captured using responsibilities described in terms of obligations, liabilities and norms. It shows how human/organisational agents are represented as being accountable for their behaviour, how resource and causal dependencies are captured and represented. And how dependencies between system-level behaviour and component level behaviour is represented using a causal map.

The example SoS, 'MegaFileShare', consists of a web server running on a virtual machine that is independently managed and operated by 'Website Host Ltd' and a 'file storage as a service' system, similar to Amazon S3, which is independently managed and operated by 'Simply Storage Ltd'. The website itself is designed and maintained by 'MegaFileShare Ltd' and users of the site upload its content, normally in the form of videos or eBooks.

The desired behaviour of this SoS is described using the responsibility abstraction in terms of obligations, liabilities and norms. Tables 4.1 and 4.2 describe a partial set of the obligations, liabilities and norms of the storage system provider, which we call 'Simply Storage Ltd', and the Website host, which we call 'Website Host Ltd'. We can see that Tables 4.1 and 4.2 are partial since the responsibilities with respect to abiding by specific laws and legislation have not been listed for simplicity. We could however specify these should it be desired but for illustrative purposes this intricacy is unnecessary.

Table 4.1 –Description of responsibilities between Simply Storage and Website Host

	Simply Storage Ltd.			Website Host Ltd.		
#	Obligation	Liability	Norms	Obligation	Liability	Norms
1	Write received files to storage	If Uptime <99.9% then refund 10% of customer service credit. If Uptime <99% then refund 25% if customer service credit. If availability is impacted by factors other than those used in our calculation of the error rate,	1. Uptime is measured on availability of service within the S3 demarcation area. Downtime due to problems outside the demarcation area such as internet issues are not included.	Pay for file storage @ $0.14 per GB	If payments are not received then liable for legal action due to breach of contract and account will be suspended until payment is	Payment taken by credit card on monthly basis.

		we may issue a Service Credit considering such factors at our sole discretion	2. Data will be stored on an infrastructure with 99.99999999 % 'durability' over a year.		received.	
2	Read requested file from storage	See above	See above	Pay for data transfer @ $.012 per GB	See above	See above

Table 4.2 – Description of Responsibilities between Website Host and MegaFileShare

#	Website Host Ltd.			MegaFileShare Ltd		
#	Obligation	Liability	Norms	Obligation	Liability	Norms
3	Provision web server and file storage	If uptime < 99.9% then financially liable for the resultant lost advertisement revenue during periods of unavailability.	Uptime is measured on availability of the service (including file download and upload) within hosting demarcation area including the internet gateway.	Pay for provision of web servers and file storage @ $2.64 per server per hour, $.12 per GB transmitted, $.14 per GB stored.	If payments are not received that liability for legal action due to breach of contract and their account will be suspended until payment is received.	Payment taken by credit card on monthly basis.

Table 4.1 and 4.2 describe the desired behaviour of the SoS. We can observe that 'Simply Storage Ltd' is responsible for writing files to storage, reading files from storage and to provide this behaviour with an availability of 99.9% during a billing period. It is specified in the table that if this behaviour is not met then refunds are owed to the customer depending on the severity of the breach. It is also specified in the table that the files should be stored on infrastructure with a 'durability' of 99.99999999%, meaning over the lifetime of the only one in every billion files may be lost/corrupted. We can also see that in return for this desired behaviour 'Website Host Ltd' is obligated to pay for file storage and for data transfer and that they must pay on a monthly basis using a credit card. Additionally it is specified that if this is not met then 'Website Host Ltd' is liable to be sued for breach of contract and their account will be suspended until payment is received.

In Table 4.2 we can see the responsibilities between 'Website Host Ltd' and 'MegaFileShare Ltd'. 'Website Host Ltd' should provide a web server and file storage that that has an uptime over 99.9%. If this desired behaviour is not met then 'Website Host Ltd' is financially liable for the resultant lost advertisement revenue during periods of unavailability. In return for this behaviour 'MegaFileShare Ltd' must pay for the services at a specified rate on a monthly basis. If they do not do so then 'MegaFileShare Ltd' is liable to be sued for breach of contract.

Having captured a specification of desirable and undesirable behaviour and the consequences of this behaviour in terms of responsibilities (obligations, liabilities and norms), one may now attempt to identify resources and interactions that produce the behaviours that have been specified. To do this one needs to identify the entities that constrain/enable the behaviours that the agents are interested in. One should first focus on those entities that contribute to fulfilling obligations that stakeholders are interested in. The most important class of entity to model are the resources that enable the responsible agent to fulfil their obligations in accordance with the norms specified. The most important class of relationship to model is the fulfilment dependency between responsibilities. A fulfilment dependency represents that for a responsibility to be fulfilled it is dependent upon the fulfilment of another. Table 4.3 below specifies the resources that are required for each of the stakeholders to fulfil their obligations as specified in Tables 4.1 and 4.2. We can see, from Table 4.3, that the file storage service provider requires a storage system with 99.9% availability and a durability of 99.999999999% in addition to an Internet gateway that provides an availability of 99.9%. We can also see that 'Website Host Ltd' requires a credit card for payment and a web server (or set of web servers) with an availability of 99.9%.

Table 4.3 - Partial description of the resources required to meet responsibilities within 'MegaFileShare' SoS

Rsp	Obligation	Resource Requirement	Obligation	Resource Requirement
1	Write file to storage	-Storage system with 99.9% uptime and durability of 99.999999999% -Internet gateway with uptime of 99.9%	Pay for file storage @ $0.14 per GB	-Credit card for payment
2	Read requested file from storage	See above	Pay for data transfer @ $.012 per GB	See above
3	Provision web server and file storage	-Web server with 99.9% uptime -File storage system with 99.9% uptime -Internet gateway with uptime of 99.9%	Pay for provision of web servers and file storage @ $2.64 per server per hour, $.12 per GB transmitted, $.14 per GB	-Credit card for payment

			stored.	

Once the responsibility model has been completed the behaviour of the system must be bounded to determine if it provides the behaviour that agents are responsible for. To bound the behaviour of a SoS, and thus identify potential behaviours that threaten the system's desired behavioural properties, one should consider how the failure of specific agents, or combinations of agents, to meet their obligations will affect the fulfilment of other obligations. A diagram of the relationships between agents, responsibilities and resources can help achieve this by encouraging analysis to be performed in a systematic way.

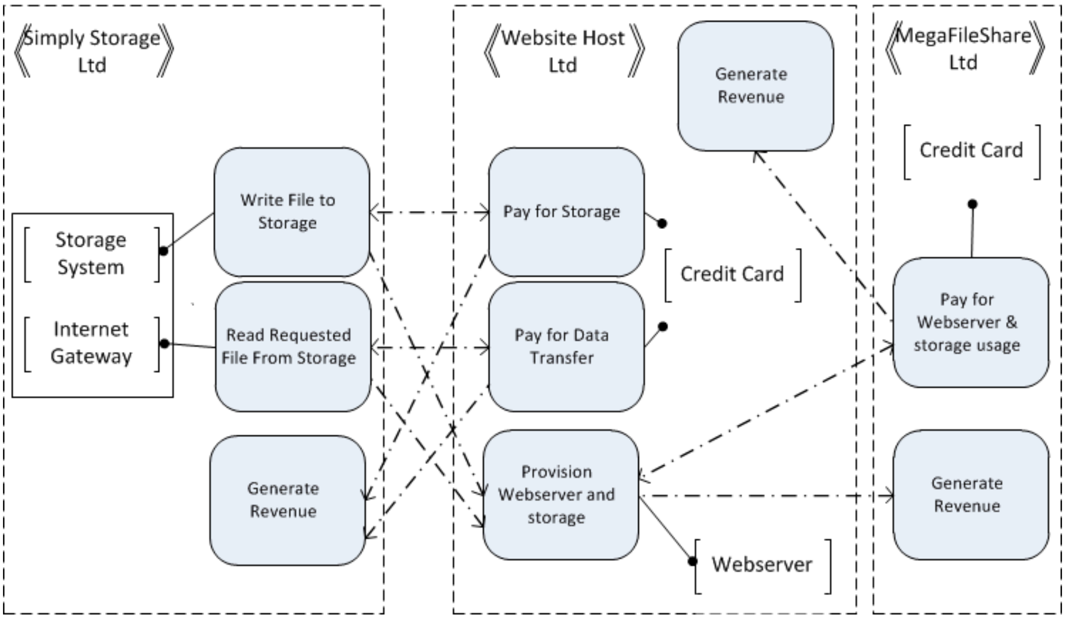


Figure 4.1 - Responsibility Model Of MegaFileShare SoS

The responsibility model above (Figure 4.1) simply describes the relationships between agents, responsibilities and resources as described in Tables 4.1, 4.2 and 4.3. We can see that 'Simply Storage Ltd' is responsible for file storage and file retrieval, 'Website Host Ltd' is responsible for paying for storage, paying for data transfer and provisioning the web server and storage for 'MegaFileShare Ltd'. We can see that the write file to storage and read requested file from storage responsibilities are dependent on two resources (the storage system and internet gateway). We can see that payment responsibilities are dependent on resources (credit cards). We can also see that there are multiple fulfilment dependencies between the responsibilities. For instance, write file to storage is dependent upon 'Website Host Ltd' paying for storage at the end of every month. Similarly 'Website Host Ltd' paying for storage is dependent upon 'Simply Storage Ltd' providing file storage the previous month.

To illustrate that we may capture the organisational agents commercial interests, the 'Generate Revenue' responsibility of each organisational agent has also been included to highlight the interdependencies between the agents' commercial interests. Hence the model describes that 'Simply Storage Ltd' may only fulfil it responsible to generate revenue if 'Website Host Ltd' meet their responsibility of paying for storage and data transfer. Similarly 'Website Host Ltd' may only fulfil their responsibility to generate revenue if 'MegaFileShare Ltd' fulfil their responsibility of paying for server provisioning and storage, and 'MegaFileShare Ltd' may only generate revenue if 'Website Host Ltd' fulfil their responsibility to provision a website and storage.

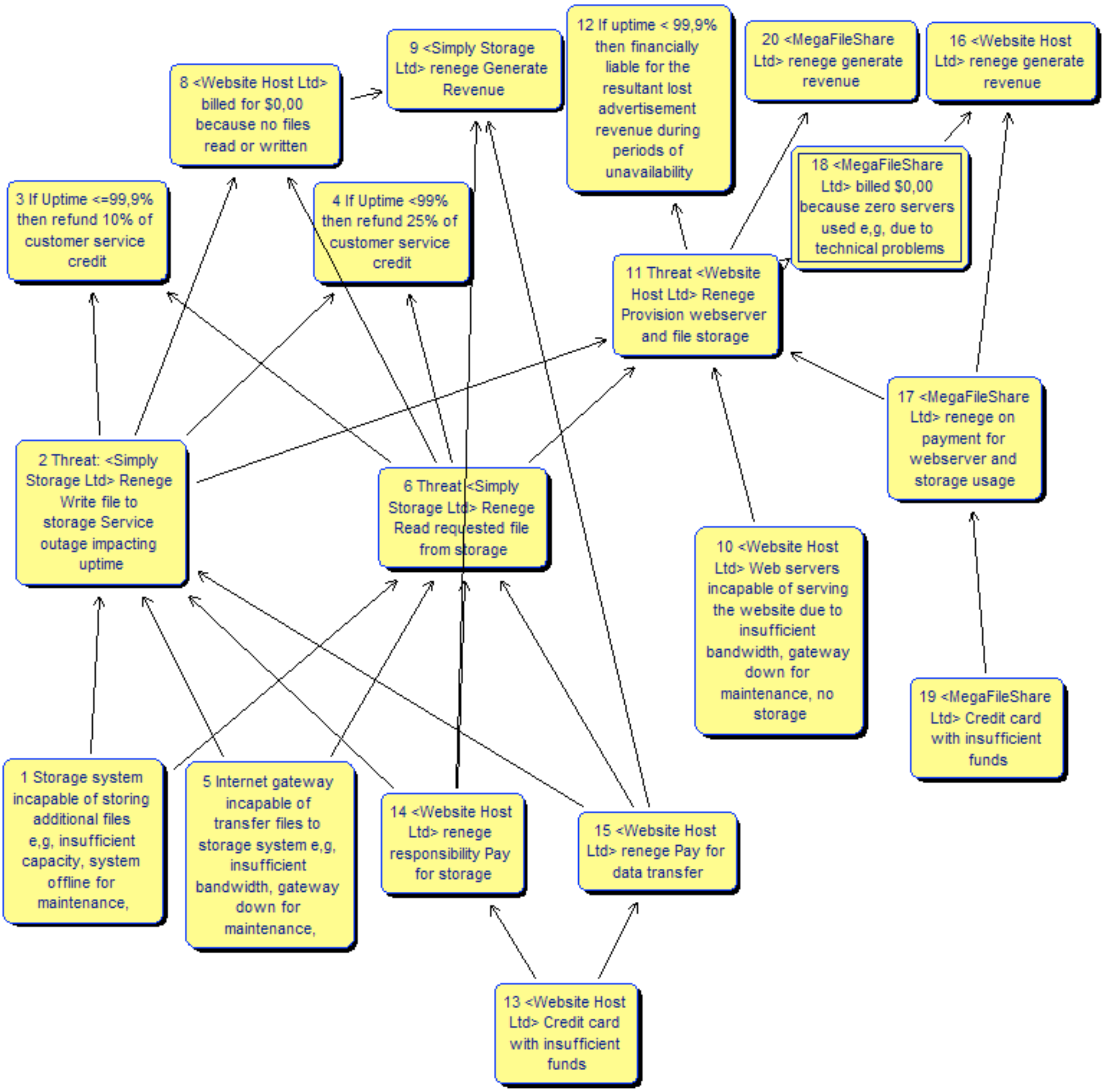
Having prepared a graphical responsibility model, threats to the desired behaviour of the system can be captured using 'risk clauses'. A risk clause is composed of a *target*, a *condition*, a set of *consequences* and an estimate of their *severity* and *likelihood*. A target is the entity or relationship to which the risk clause refers. Typically the target of a risk clause will be a responsibility. A condition is a description of a particular state that could manifest to threaten the target, e.g. a lack of appropriate resources to fulfil an obligation. The consequences describe the effects of the condition on the target e.g. the responsibility will be reneged until adequate resources are obtained. The severity describes the liabilities resulting from the threat manifesting e.g. that the agent faces a financial penalty for reneging the responsibility. Table 4.4 below describes the risk clauses associated with the 'MegaFileShare'.

Table 4.4 - Risk Clauses for 'MegaFileShare'

Target	Condition	Consequences	Severity
<Simply Storage Ltd> Write received file to storage	<1> Storage system incapable of storing additional files e.g. insufficient capacity, system offline for maintenance, … <5> Internet gateway incapable of transfer files to storage system e.g. insufficient bandwidth, gateway down for maintenance, …	<2> Service outage impacting uptime. -Lost revenue from downtime. <3> -If Uptime <99.9% then refund 10% of customer service credit. <4> -If Uptime <99% then refund 25% of customer service credit.	Severity dependent on duration of down time. -Larger the down time larger the refund <8> WebsiteHost Ltd billed for $0.00 because no files written <9> Renege Generate Revenue
<Simply Storage Ltd> Read requested file from storage	<1> Storage system incapable of retrieving files e.g. file corruption, insufficient IO bandwidth to retrieve file, system offline for maintenance, … <5> Internet gateway incapable of transfer files to storage system e.g. insufficient bandwidth, gateway down for maintenance, …	<6> Service outage impacting uptime. -Lost revenue from downtime. <3> -If Uptime <=99.9% then refund 10% of customer service credit. <4> -If Uptime <99% then refund 25% if customer service credit.	Severity dependent on duration of down time. -Larger the down time larger the refund <8> WebsiteHost Ltd billed for $0.00 because no files written <9> Renege Generate Revenue
<Simply Storage Ltd>Generate Revenue	<14> <Website Host Ltd> renege responsibility to pay for Storage used <8> <Website Host Ltd> billed for $0.00 because no files read or written e.g. technical problems	<9> Renege Generate Revenue	Severity dependent on the duration.
<WebsiteHost Ltd>	<10> Web servers incapable of serving the website e.g.	<12> If uptime < 99.9% then financially liable for the	High severity

Provision web server and file storage	insufficient bandwidth, gateway down for maintenance, … <2> <Simply Storage Ltd> renege responsibility to write file to storage <6> <Simply Storage Ltd> renege responsibility to read requested file from storage	resultant lost advertisement revenue during periods of unavailability. <18> <MegaFileShare> billed $0.00 because service not provisioned	<20> <MegaFileShare> renege generate revenue. Website host financially liable for lost revenue of ‘MegaFileShare Ltd’. <16> <Website Host> renege generated revenue
<Website Host Ltd> Pay for storage	<13> 1. Credit card with insufficient funds	<14> Account is suspended until payment is received. If payments are not received then liable for legal action due to breach of contract and their	Extreme severity <2><6>No longer able to fulfil responsibility to provision web servers and file storage
<Website Host Ltd> Pay for data transfer	As above	<15> As above	As above
<Website Host Ltd> Generate revenue	<17> <MegaFileShare Ltd> renege on payment for web server and storage usage <18> <MegaFileShare Ltd> billed $0.00 because zero servers used e.g. due to technical problems	<16> <Website Host Ltd> Renege generate revenue	None specified
<MegaFileShare Ltd> Pay for provision of servers and file storage	<19> <MegaFileShare Ltd> Credit card with insufficient funds	<17> <MegaFileShare Ltd> renege on payment for webserver and storage usage. <11> <Website Host Ltd> Account is suspended until payment is received. If payments are not received then liable for legal action due to breach of contract and their	Extreme severity <16> <Website Host Ltd> No longer able to generate revenue since web server and storage will not be provisioned until payment received <20> <MegaFileShare> Renege generate revenue
<MegaFileShare Ltd> Generate Revenue	<11> <Website Host Ltd> renege on provisioning web server and storage	<18> <MegaFileShare> billed $0.00 because service not provisioned	Extreme severity <20> <MegaFileShare> Renege generate revenue <16> <Website Host Ltd> No longer able to generate revenue since web server and storage will not be provisioned until payment received

The above risk clauses can be made more easily interpretable by representing them as a causal map. This assists in the identification of relationships between clauses hence it enables the representation of relationships between specific resource conditions or agent conditions to desired system level behaviours as may be represented using responsibilities. Below is the causal map of the risk clauses in Table 4.4.



8 <Website Host Ltd> billed for $0,00 because no files read or written
9 <Simply Storage Ltd> renege Generate Revenue
12 If uptime < 99,9% then financially liable for the resultant lost advertisement revenue during periods of unavailability
20 <MegaFileShare Ltd> renege generate revenue
16 <Website Host Ltd> renege generate revenue
18 <MegaFileShare Ltd> billed $0,00 because zero servers used e,g, due to technical problems
3 If Uptime <=99,9% then refund 10% of customer service credit
4 If Uptime <99% then refund 25% of customer service credit
11 Threat <Website Host Ltd> Renege Provision webserver and file storage
17 <MegaFileShare Ltd> renege on payment for webserver and storage usage
2 Threat: <Simply Storage Ltd> Renege Write file to storage Service outage impacting uptime
6 Threat <Simply Storage Ltd> Renege Read requested file from storage
10 <Website Host Ltd> Web servers incapable of serving the website due to insufficient bandwidth, gateway down for maintenance, no storage
19 <MegaFileShare Ltd> Credit card with insufficient funds
1 Storage system incapable of storing additional files e,g, insufficient capacity, system offline for maintenance,
5 Internet gateway incapable of transfer files to storage system e,g, insufficient bandwidth, gateway down for maintenance,
14 <Website Host Ltd> renege responsibility Pay for storage
15 <Website Host Ltd> renege Pay for data transfer
13 <Website Host Ltd> Credit card with insufficient funds

Figure 4.2 - Causal map of Risk Clauses

By inspecting the causal map of the risk clauses one may identify the relations between risk clauses. One can identify the responsibilities that are reneged as a result of certain conditions by following the arrows from condition up to a reneged responsibility. One can identify the conditions that cause a responsibility being reneged by following the arrows in reverse thus enabling threat identification and troubleshooting using the diagram. For instance we can identify that an insufficiency of storage resources in ‘Simply Storage Ltd’ storage system is a threat to ‘Website Host Ltd’s responsibility to generate revenue. Equally we can identify that the threats to ‘MegaFileShare Ltd’ generating revenue include ‘Website Host Ltd’ having insufficient funds on their credit card resulting in a suspension of the storage system, the ‘Simply Storage Ltd’ storage system becoming incapable to meet demand, the ‘Simply Storage Ltd’ internet gateway becoming incapable to meet demand, the ‘Website Host Ltd’ web servers being incapable to meet demand, or their own credit card having insufficient funds pay for ‘Website Host Ltd’ services.

In later chapters of this thesis, we will develop tactics for identifying ‘socio’ causes of undesirable behaviour by illustrating how it may be possible to identify overlapping self-interests (or otherwise) by inspection of obligations and liabilities, and agents’ potential to conflict/resist a system.

4.5 Why Responsibilities are a Useful Abstraction

Responsibilities are a useful abstraction for socio-technical threat analysis and troubleshooting as they enable the modelling of systems in manner that makes the desires of a stakeholder and the importance of each desire a focal point of analysis. For instance, the website owner of ‘MegaFileShare’ wanted a file sharing website with 99.9% uptime, and this behavioural property is worth making the web host liable for the value of the revenue lost for the duration of downtime.

Responsibilities are also useful because they present a system in manner that makes accountability a focal point of analysis. This is particularly important for threat identification and troubleshooting across organisational boundaries as it enables the analysis of who is ultimately liable for what, and therefore can aid in understanding whether threats/risks have been transferred to partners or otherwise.

Responsibilities are also useful because their capture of norms can help in the analysis of misunderstanding when working with agents across organisational boundaries. For instance the way ‘Uptime’ is calculated by the storage service provider may be different to what another agent believes is the norm. However by capturing this explicitly as norm it is brought to the fore.

Responsibilities are also useful for practical reasons. They are straightforward to elicit as they are a concept that is in common verbal use. It is also relatively trivial to ask stakeholders about their norms. For example, how is this normally done? How would you expect this to be done? Responsibilities are also particularly useful for representing large systems as they can be used to abstract away task/process level information.

Perhaps most importantly responsibilities ground the model on important mediators of social action e.g. obligations, liabilities norms. We know from behavioural studies that human behaviour is strongly related to a human's beliefs about the *consequences of behaviour*, *social normative pressures* and the *presence of factors that may facilitate* or impede performance (Ajzen, 1991; Armitage & Conner, 2001). Responsibility modelling captures these notions as the *consequences of behaviour* can be captured using obligations and liabilities to detail benefits or sanctions associated with performance and non-performance of behaviour. *Social normative pressures* can be captured by using norms, and *presence of facilitating factors* can be captured using the idea of resource availability and being dependent on others fulfilling their interdependent responsibilities.

4.6 A Comparison with other Modelling Abstractions

In this section we can consider the extent that three prominent systems modelling approaches (IDEF, UML, I*) and responsibility modelling meet the requirements of an abstraction for socio-technical troubleshooting and threat analysis.

Our findings are that responsibility modelling meets all of the information and practical requirements. IDEF meets four of the six information requirements and both practical requirements. UML meets three of the six information requirements and both practical requirements. And I* modelling meets all six of the information requirements but meets only one of the two practical requirements. These findings are summarised in Table 4.5 below and explained in more detail in the relevant sections below.

Table 4.5 - Summary of mapping between requirements and systems modelling approaches

Requirement	IDEF	UML	I*	Responsibility
1-A desirable and undesirable behaviour;	**Yes – satisfied by IDEF0 functions and control**	**No – requires some modification to use case to capture undesirable behaviour e.g. 'non-use cases'.**	**Yes – satisfied by use of goal in a strategic dependency model**	**Yes – satisfied by use of responsibilities and norms**
1-B the consequences of undesirable behaviour for a *component* and its *interdependent components*;	**Yes – satisfied by IDEF3 transition schematics**	**Yes – satisfied using sequence diagrams**	**Yes – satisfied by use of goal, resource, and task dependencies in a strategic dependency model**	**Yes – satisfied by use of Agent liabilities and Risk clauses.**
1-C the consequences of desirable behaviour not occurring for a *component* and its *interdependent components*;	**Yes – satisfied by IDEF3 transition schematics**	**Yes – satisfied using sequence diagrams**	**Yes – satisfied by use of goal, resource, and task dependencies in a strategic dependency model**	**Yes – satisfied by use of Agent liabilities and risk clauses.**
1-D the component that is accountable for the (non)occurrence of behaviour;	**No - requires some modification to semantics so that accountability is representable by associating functions with human mechanisms or by adding an additional attribute to functions**	**No – requires some imaginative use of use case pre and post conditions to prescribe accountability**	**Yes – satisfied by associating a goal with an agent in a strategic dependency model**	**Yes – satisfied by associating responsibilities to agents using the 'responsible for' relation**
1-E the resources that a component depends on to perform behaviour;	**Yes – satisfied by IDEF0 dependencies represented as functions inputs, mechanisms and controls**	**Yes – satisfied by dependencies in class and sequence diagrams**	**Yes –satisfied using strategic rational resource dependencies**	**Yes – satisfied dependencies using resource dependencies**
1-F the dependencies between system-level behaviour and component level behaviour.	**No – requires derivation by tracing relations between functions using IDEF3 transition schematics and IDEF0 models.**	**No – requires derivation by tracing relations between use cases and sequence diagrams**	**Yes – satisfied using hierarchies of system level goals and component level goals**	**Yes – satisfied using causal map of risk clauses that represents dependencies between system level behaviour and component level behaviour**

4.6.1 IDEF

A combination of IDEF0 and IDEF3 may be used to meet four (1-A, 1-B, 1-C, 1-E) of the six information requirements and meet both of the practical requirements (2, 3). With some modification IDEF could meet all six information requirements.

IDEF meets Requirement 1-A the capability to capture desirable and undesirable behaviour. The requirement is met using IDEF0's capability to capture desirable behaviour using functions and its capability to capture undesirable behaviour by specifying controls on functions.

IDEF meets Requirement 1-B the capability to capture the consequences of undesirable behaviour for a component and its interdependent components. The requirement is met using IDEF3 transition schematics that may be used to describe consequences of actions on interdependent components.

IDEF meets Requirement 1-C the capability to capture the consequences of desirable behaviour not occurring for a component and its interdependent components. The requirement is met using IDEF3 transition schematics that represent behavioural interactions between components.

IDEF does not meet Requirement 1-D the capability to capture the component that is accountable for the (non)occurrence of behaviour. IDEF may be modified to met this capability by adding additional semantics to IDEF0 models so that humans acting as 'mechanisms' are deemed accountable for that function. Alternatively an additional 'responsibility of' attribute could be added to functions to capture this information.

IDEF meets Requirement 1-E the capability to capture the resources that a component depends on to perform behaviour. This requirement is met using IDEF0 models that capture inputs, controls and mechanisms. Resources may be represented as either: inputs of functions (if they are consumed); controls (if they mediate the conversion of inputs to outputs), or mechanisms that perform the conversion or aid in the conversion.

IDEF does not meet Requirement 1-F the capability to capture dependencies between system-level behaviour and component level behaviour. IDEF provides no concise representation of these dependencies however dependencies between system level behaviour and component level behaviour may be derived by manually tracing component level functions up the IDEF0 functional hierarchies to identify their impact on higher-level functions.

IDEF meets Requirement 2 'the model should have the potential to represent systems with non-trivial numbers of components and interactions'. IDEF enables the representation of large systems using hierarchies of models of interdependent functions, controls, mechanisms, calls, inputs and outputs.

IDEF meets Requirement 3 'the analysis of the models should have the capability to be algorithmic, or computer assisted, such that the scale of an analysable system is not limited by the number of components a person can interpret'. IDEF3 models support computer-aided analysis by being converted into stocks and flows simulations using tools such as PROSIM[8] or IBM Rational System Architect[9].

4.6.2 UML

A combination of use cases, class diagrams and sequence diagrams, may be used to meet three (1-B, 1-C, 1-E) of the six information requirements and meet both of the practical requirements (2, 3). With some modification UML could meet all six information requirements.

UML do not meet Requirement 1-A the capability to capture desirable and undesirable behaviour. Desirable behaviour maybe captured using use case diagrams however capturing undesirable behaviour is not supported as standard. UML use cases could be extended to support 'non-use case' to meet the requirement. Alternatively high-level sequence diagrams could be used to capture high-level desired and undesired behaviour.

UML meets Requirement 1-B the capability to capture the consequences of undesirable behaviour for a component and its interdependent components. Sequence diagrams may be used to capture consequences of undesirable behaviour.

UML meets Requirement 1-C the capability to capture the consequences of desirable behaviour not occurring for a component and its interdependent components. Sequence diagrams may be used to capture consequences of non-occurrence of desirable behaviour.

UML does not meet Requirement 1-D the capability to capture the component that is accountable for the (non)occurrence of behaviour without some adjustments. With some imagination accountability may be included as pre and post-conditions of use-cases. Otherwise class diagrams could be modified to include attributes linking accountable persons to component behaviours.

UML meets Requirement 1-E the capability to capture the resources that a component depends on to perform behaviour. This is met using class models and sequence diagrams that capture interdependent objects and actors.

UML does not meet Requirement 1-F the capability to capture dependencies between system-level behaviour and component level behaviour. UML provides no concise representation of the relationship between system behaviours and component behaviours. However system level behaviour and component level behaviour may be traced by manually tracing which object interactions implement which use cases thereby linking system behaviours with component level behaviours.

[8] http://www.kbsi.com/COTS/ProSim.htm

[9] http://www-01.ibm.com/software/awdtools/systemarchitect/features/simulation.html

UML meets Requirement 2 'the model should have the potential to represent systems with non-trivial numbers of components and interactions'. UML enables the representation of large systems using hierarchies of models that provide perspectives of different granularity.

UML meets Requirement 3 'the analysis of the models should have the capability to be algorithmic, or computer assisted, such that the scale of an analysable system is not limited by the number of components a person can interpret'. Sequence diagrams support computer-aided analysis by being converted into stocks and flows simulations using tools such as IBM Rational Software Architect[10] or by being converted into Petri nets or queuing models (King & Pooley, 2000; Li & Yao, 2009; Pooley & King, 1999).

4.6.3 I*

I* strategic rationale models may be used to meet all six (1-A, 1-B, 1-C, 1-D, 1-E, 1-F) of the information requirements but meet only one of the two practical requirements (3). With some modification strategic rationale models could meet all six information requirements and both practical requirements.

Strategic rationale models meet Requirement 1-A the capability to capture desirable and undesirable behaviour. Desirable behaviour maybe captured as a goal or soft goal. Undesirable behaviour may be captured as a goal or soft goal to avoid/prevent an undesirable behaviour or state.

Strategic rationale models meet Requirement 1-B the capability to capture the consequences of undesirable behaviour for a component and its interdependent components. This is because the consequences of an agent's behaviour on themselves are captured in terms its impact on their goals. The consequences on interdependent agents are also captured through its impact on other agents' goals.

Strategic rationale models meet Requirement 1-C the capability to capture the consequences of desirable behaviour not occurring for a component and its interdependent components. It is met because strategic dependencies capture the consequences of goals being satisfied or otherwise.

Strategic rationale models meet Requirement 1-D the capability to capture the component that is accountable for the (non)occurrence of behaviour without some adjustments. Accountability may be captured by claiming that agents are accountable for the goals that are associated to them.

Strategic rationale models meet Requirement 1-E the capability to capture the resources that a component depends on to perform behaviour. This is met using the notion of resource dependencies.

[10] http://www-01.ibm.com/software/rational/products/swarchitect/simulation/

Strategic rationale models meet Requirement 1-F the capability to capture dependencies between system-level behaviour and component level behaviour. This is because dependencies between system level behaviour and component level behaviour are represented using hierarchies of goals.

Strategic rationale models do not meet Requirement 2 'the model should have the potential to represent systems with non-trivial numbers of components and interactions'. In practice, I* models are acknowledged to be limited in size due to a lack of model hierarchies which results in models that are difficult to comprehend and manage at scale. This limitation may be rectifiable through the use of viewpoints or other ways of hiding information that is irrelevant for a certain purpose.

Strategic rationale models meet Requirement 3 'the analysis of the models should have the capability to be algorithmic, or computer assisted, such that the scale of an analysable system is not limited by the number of components a person can interpret'. Some computer-aided analysis is provided to analyse goal models and assess whether goals are satisfied (Horkoff & Yu, 2009, 2010).

Overall, the findings mean that whilst the *analysis* of I* strategic rationale model do *scale* due to the scalability of computer-aided analysis, the graphical representations do not scale as they do not currently support hierarchies of models, viewpoints or other ways of representing large volumes of modelling data.

4.6.4 Responsibility models

Responsibility models may be used to meet all six (1-A, 1-B, 1-C, 1-D, 1-E, 1-F) of the information requirements and meet both of the practical requirements.

Responsibility models meet Requirement 1-A the capability to capture desirable and undesirable behaviour. Desirable and undesirable behaviour may be represented as responsibilities to attain, maintain or avoid a given state. Desirable / undesirable ways of attaining given states can be specified as norms.

Responsibility models meet Requirement 1-B the capability to capture the consequences of undesirable behaviour for a component and its interdependent components. The consequences of undesirable behaviour (of a component) may be captured using the notion of a liability. The consequences of undesirable behaviour on interdependent components may be captured using the notion of a risk clause.

Responsibility models meet Requirement 1-C the capability to capture the consequences of desirable behaviour not occurring for a component and its interdependent components. The consequences of desirable behaviour (of a component) not occurring may be captured using the notion of a liability. The consequences of desirable behaviour not occurring on interdependent components may be captured using the notion of a risk clause.

Responsibility models meet Requirement 1-D the capability to capture the component that is accountable for the (non)occurrence of behaviour. Accountability is represented by associating responsibilities to agents using a 'responsible for' relation.

Responsibility models meet Requirement 1-E the capability to capture the resources that a component depends on to perform behaviour. This is met using the notion of resource dependencies.

Responsibility models meet Requirement 1-F the capability to capture dependencies between system-level behaviour and component level behaviour. Dependencies between system level behaviour and component level behaviour may be represented using a causal map of risk clauses.

Responsibility models meet Requirement 2 'the model should have the potential to represent systems with non-trivial numbers of components and interactions'. Responsibility models enable the representation of large system by using hierarchies of models and viewpoints. For instance (Ramduny-Ellis & Dix, 2007) use hierarchies of models to represent and analyse the responsibilities associated with book production and represent a high-level responsibility by breaking it down into sub-models that represent the responsibilities associated with book planning, entering into a contract, producing the text and transferring ownership to the publisher. Another example is the work of (Dobson, 2007) that uses differing viewpoints of a system to identify inconsistencies between the ways responsibilities are allocated for different operational tasks such as project management vs. quality management, and project management vs. system development.

Responsibility models meet Requirement 3 'the analysis of the models should have the capability to be algorithmic, or computer assisted, such that the scale of an analysable system is not limited by the number of components a person can interpret'. Basic computer aided analysis supports identifying unassigned responsibilities and finding agents that are overloaded with responsibilities (Lock et al., 2009). Tools for computer-aided analysis of causal maps such as Banxia Decision Explorer[11] are also available (Montibeller & Belton, 2006).

In summary, responsibility modelling is a promising abstraction for meeting the requirements set out for modelling and analysing SoS for the purpose of threat analysis and troubleshooting. In subsequent chapters the responsibility abstraction is applied to real-world case studies to demonstrate its efficacy in practice. In the next section this claim is illustrated by means of a case study that shows that the responsibility abstraction may be used to identify threats to the dependability of a system.

[11] http://www.banxia.com/dexplore/

4.7 Chapter summary

This chapter demonstrated the feasibility of using responsibility modelling as an abstraction for SoS threat identification and troubleshooting. It was argued that whilst creating and maintaining a descriptively complete structural and behavioural model of SoS is unrealistic, the more modest strategy of creating models that *bound* the possible behaviour of a SoS may be realistic. It was proposed that high-level structural models might be sufficiently invariant and informative to enable analysis that bounds the behaviour of a SoS. It was then proposed that a model should capture six classes of information to enable the socio-technical threat identification and troubleshooting of a SoS. These consisted of the capture of:

1. desirable and undesirable behaviour;
2. the consequences of undesirable behaviour for a component and its interdependent components;
3. the consequences of desirable behaviour not occurring for a component and its interdependent components;
4. the person/organisation that is accountable for the (non)occurrence of behaviour;
5. the resources that a component depends on to perform the behaviour that is accountable for;
6. the dependencies between system-level behaviour and component level behaviour.

It was then demonstrated using an example SoS, called ‘MegaFileShare’, that responsibility modelling can be used to capture these six classes of information and thus be used for SoS threat identification and troubleshooting.

Furthermore, it was explained that responsibility modelling is a promising abstraction for socio-technical troubleshooting and threat identification because of its stakeholder centric focal point, I.e. it focuses on desirable and undesirable behaviours and their *contextualised* importance to a stakeholder. It analyses which agents (stakeholders) are ultimately liable for system behaviours, and thus enables stakeholders in boundary crossing systems to understand the risks they are accepting and those that they are transferring to other agents. Another important dimension was that the analysis of norms can be used to verify the way responsibilities are to be fulfilled and whether this meets the stakeholder’s expectations.

It was also explained that responsibility modelling is of practical use because responsibilities tend to be unproblematic to elicit and that the notions used in responsibility modelling are grounded on notions used in the behavioural sciences to predict human behaviour and thus may enable the analysis of ‘socio’ issues.

Finally, the responsibility modelling abstraction was compared against the prominent modelling abstractions UML, IDEF and I*. It was observed that the responsibility modelling abstraction was the only abstraction to meet all six information requirements and practical requirements without significant alteration of the abstraction. The affordances of the others abstractions were also discussed.

5. Responsibility Modelling for Threat Identification

5.1 Introduction

The aim of this chapter is to demonstrate the use of responsibility modelling for threat identification and risk analysis. What I mean by ‘threat identification’ is the identification of configurations of responsibilities, agents, and resources that may lead to system behaviours that are deemed undesirable by stakeholders. What I mean by ‘risk analysis’ is the assessment of the likelihood and impact of these threats on behavioural properties, or system behaviours, which are of interest to stakeholders. In practice this may mean, the identification of configurations of responsibilities, agents, and resources that lead to a system failing to meet a desired level of dependability or a specific functional requirement.

To illustrate the use of responsibility modelling for threat identification and risk analysis this chapter provides a case study of the risk assessment of a cloud computing based system-of-systems (SoS). This SoS is a particularly interesting system, as unlike some classes of SoS, its ongoing operation is dependent upon the organisational agents’ overlap of self-interests. We call this subclass of SoS, a coalition-of-systems (CoS) to emphasise the vital role of the overlap of agents’ interests in the ongoing operation of the system. The fact that the system is a CoS makes the analysis of this system’s dependability particularly interesting. This is because the fragility of the coalition’s overlapping self-interests needs to be analysed to understand the system’s ongoing behaviour. I will demonstrate how this may be achieved.

In the next chapter I further illustrate the use of responsibility modelling for threat identification and risk analysis by demonstrating how it can be used to identify threats arising from a system *conflicting* with agents’ interests.

5.2 Coalitions of Systems and Socio-Technical Threat Identification

System-of-Systems (SoS) are a class of system whose interacting parts comprise systems, that are owned and managed by independent parties, and whose parts evolve over time (Maier, 1998). Typical examples are integrated supply chain management systems, integrated healthcare networks, and cyber-physical systems such as integrated embedded systems within ships, land vehicles, aircraft, or industrial plants. Coalitions-of-systems (CoS) are a sub-class of SoS that have the additional property that their subsystems interact to further overlapping self-interests rather than achieving an overarching mission/goal as in the case of typical SoS. The distinction between SoS and CoS is useful when conducting a socio-technical analysis as it indicates whether the alignment of partners interests needs to be analysed. When analysing commercial SoS this distinction is often required as organisations are driven by profit and the rewards/risks will be distributed between partners so understanding these interests become important to understanding its ongoing operation.

The dependability of a system, in this context, is definable as the property of a system where “reliance can justifiably be placed on the services it delivers” (Sommerville, Dewsbury, Clarke, & Rouncefield, 2006). We define threats to dependability as events or conditions that affect a systems availability, reliability, safety, integrity, confidentiality, and maintainability (Avizienis, Laprie, Randell, & Landwehr, 2004; Sommerville et al., 2006).

Socio-technical threats are an important factor when analysing the dependability of a CoS. This is because coalition partners overlapping self-interests may be fragile and subject to change. For example, a change in a cloud provider’s interests may result in the withdrawal of certain services, or changes to their behavioural properties, thus resulting in threats to the availability, reliability and maintainability of the overall CoS. Understanding the distribution of liabilities among coalition partners is also important to understanding CoS as this provides an indicator of consequences of a partner’s action and its implications for their interests. For instance, in resource constrained situations coalition partners that seek to further their self-interest by generating profit will fulfil responsibilities with large liabilities and renege on those where the liability is small.

There are a number of candidate socio-technical modelling approaches relevant to identifying and assessing dependability threats. Recently there has been a trend for agent goal model based frameworks such as I* (E. Yu et al., 2011) and the TROPOS Goal-Risk framework (Asnar & Giorgini, 2007), however this thesis advocates an agent responsibility based identification approach. Below we provide an overview of agent goal based modelling approaches prior to distinguishing them from our agent responsibility based approach (Lock et al., 2009; Sommerville et al., 2009).

The I* framework has been extended by (Maiden & Jones, 2004; Maiden et al., 2006; Mayer et al., 2007) in order to identify dependability threats and requirements for air traffic management (ATM) and enterprise systems. In the context of ATM this was achieved by means of exploring the consequences of one actor fulfilling two or more roles and whether this affects the system’s overall goal attainment (Maiden & Jones, 2004; Maiden et al., 2006). This work primarily focused on identifying system dependability related to overloading operators. In the context of enterprise systems, threats to security goals were identified by modelling and discovering business assets, constraints and security requirements. This work primarily focused on gathering security requirements and transforming these requirements into high-level controls.

The TROPOS Goal-Risk framework was used in (Asnar & Giorgini, 2007; Asnar et al., 2008) as an approach to analyse and mitigate threats to the goal accomplishment in an ATM system and a manufacturing organisation. The systems under analysis were modelled as a configuration of related goals, tasks and events. The framework comprises a goal layer representing the goals of actors that should be achieved, an event layer that represents potential threats to goals, and a treatment layer that comprises possible threat management strategies. This approach primarily focuses on the analysis of threats and the design of appropriate high level controls rather than their identification.

Despite the agent responsibility abstraction having some similarities to goal modelling based approaches it differs significantly. Responsibility modelling uses the concept of responsible agents (human / organisational agents) and their interactions to represent a situation and identify hazards in terms of failures of agents to fulfil responsibilities. The concept of responsibility foregrounds notions of obligations, liabilities, and conformance to norms, or standards, such that it is important how an agent acts. For example, a doctor that has performed procedures in accordance with legal and domain standards may have successfully discharged their responsibility for patient care even if their patient dies. Similarly if a patient lives but their treatment was unethical then the doctor will be held accountable. Unlike responsibilities, goals principally focus on what has to be achieved.

The responsibility modelling based approach offers a number of attractive characteristics that may make it suitable for identifying threats to CoS. Firstly the agent responsibility abstraction provides a natural way of identifying the threats associated with relying on other parties to discharge responsibilities. Secondly responsibilities are relatively unproblematic to elicit as people find them ‘natural’ to articulate in comparison to ‘technical’ constructs such as functions or goals. Thirdly responsibility modelling is relatively rapid to perform as, unlike typical goal based identification approaches, tasks and their dependencies are not elicited.

The responsibility modelling approach presented here is primarily a threat/risk identification technique and should be viewed as complementary to the previous approaches discussed and more general threat/risk analysis approaches such as CORAS (Braber et al., 2003). Responsibility modelling has been used to analyse the failure of socio-technical systems including E-counting systems in the Scottish elections and UK civil emergency planning (Lock et al., 2009; Sommerville et al., 2009).

5.3 Developing Responsibility Models for Threat Identification

Responsibility-based threat identification aims to identify conditions that result in a system not exhibiting required behaviours or behavioural properties. In terms of the responsibility abstraction a system in represented as an aggregate of agents, resources and responsibilities. Required behaviours are represented by responsibilities which represent units of behaviour that an agent has a duty to perform; a responsibility model is thus a description of the behaviours that agents (within the scope of analysis) have a duty to perform and the resources they require to perform these behaviours.

As a note of caution, it is important to define the boundaries of the responsibility model so that it includes the system’s various stakeholders including those whom are the source of its behavioural requirements and those whom have to maintain or operate the system.

Responsibility models may be developed and analysed in any manner that is flexible enough to meet an analysts needs. For the purposes of performing responsibility-based threat identification a two-part process may be followed. Part 1 consists of building a model; part 2 consists of analysing the model.

5.3.1 Building a responsibility model

A responsibility model may be built using the following process:

1. Identify the focal system of the study;
2. Identify its key stakeholders (e.g. agents that consume the systems outputs, agents that provision the system);
3. Identify agents' responsibilities (obligations, liabilities, norms);
4. Identify the resources that the agents' require to fulfil their responsibilities (information, equipment);
5. Identify non-resource related dependencies the agents' require to fulfil their responsibilities (dependent upon other agents fulfilling their responsibilities);
6. Represent the information graphically using the responsibility modelling notation;

The model may be analysed once the analyst is satisfied that the elicited responsibilities (obligations, liabilities, norms) represent the behaviours required of the system. As previously described the obligation part of a responsibility specifies a unit of behaviour that an agent has duty to perform e.g. attain, maintain or avoid a given state. The liability captures the consequences, or penalty, of not fulfilling the responsibility. Norms specify the manner in which the behaviour should be performed e.g. in accordance with specific domain standards, laws or regulations.

5.3.2 Analysing a responsibility model for threats

Analysis of a responsibility model for threats may be performed by:

1. Identifying risk clauses - possible conditions that trigger events that threaten to result in a responsibility being reneged. This can be performed by:

a. inspecting the dependencies of each responsibility for *resource related threat conditions*.
e.g. insufficient/inadequate resource to fulfil responsibility.

b. inspecting the dependencies of each responsibility for *agent related threat conditions*.

e.g. insufficient time, insufficient/inadequate skill, insufficient interest to fulfil responsibility.

2. Identifying dependencies between risk clauses. This can be performed by:

a. inspecting a set of risk clauses to identify dependencies that may result from the occurrence of one risk clause triggering another.

During step 1 of analysis, HAZOPS style keywords may be used to aid in the identification of risk clauses by encouraging the analyst to systematically consider different types of condition that may threaten the fulfilment of responsibilities and thus the desired behavioural behaviour of a system.

During step 2 of analysis, causal maps can be used to aid the identification and representation of dependencies between risk clauses. To do this, the condition, threat and severity attributes of risk clauses are represented as a nodes and the belief that the condition causes the threat and the threat has consequences due to a responsibility may be represented as an arrow from one node to another. By representing all identified risk clauses in this manner one may visually inspect the causal map in a bottom up manner thus enabling the identification of threats that specific conditions trigger (including triggering cascades of threats). One may also inspect the causal map in a top-down manner to identify all the possible triggers of a responsibility being reneged.

5.4 Case Study

5.4.1 The Situation

The case study organisation is a UK based company (Company B) that provides bespoke IT solutions for the oil & gas industry. It comprises around 30 employees with offices in the UK and the Middle East. It has an organisational structure based on functional divisions (e.g. administration, engineering and support). We became involved with the organisation as a result of their interest in exploring the cost saving opportunities that cloud computing could offer them. We therefore collaborated with the organisation to assess the feasibility of migrating one of their primary service offerings (a quality monitoring and data acquisition system) to Amazon EC2 – an infrastructure-as-a-service offering from Amazon Web Services. Naturally, we were aware that the introduction of cloud technology could have adverse effects on the dependability of their service so along with cost analysis and conflict analysis (described in the next chapter), we analysed potential threats to the system's dependability as a result of its reliance upon EC2.

The situation was as follows: Company C is a small oil and gas company that owns some offshore assets in the North Sea oilfields. Company C needed a data acquisition system to allow them to manage their offshore operations by monitoring data from their assets on a minute-by-minute basis. Company C's assets rely on the production facilities of Company A (a major oil company), therefore the data comes onshore via Company A's communication links.

Company C does not have capability to develop their own IT systems; hence they outsourced the development and management of the system to Company B, which is an IT solutions company with a small data centre. The existing system was composed of two parts:

- A database server that logs and archives the data coming in from offshore into a database. A tape drive is used to take daily backups of the database, the tapes are stored off-site.
- An application server that hosts a number of data reporting and monitoring applications. The end users at Company C access these applications using a remote desktop client over the Internet.

The system infrastructure was deployed in Company B's data centre and went live in 2005. Since then, Company B's support department have been maintaining the system and solving any problems that have risen. This case study investigated the dependability threats of deploying the same system using the cloud offerings of Amazon Web Services. Figure 5.1 provides an overview of this scenario, where Company B deploys and maintains the same system in the cloud.

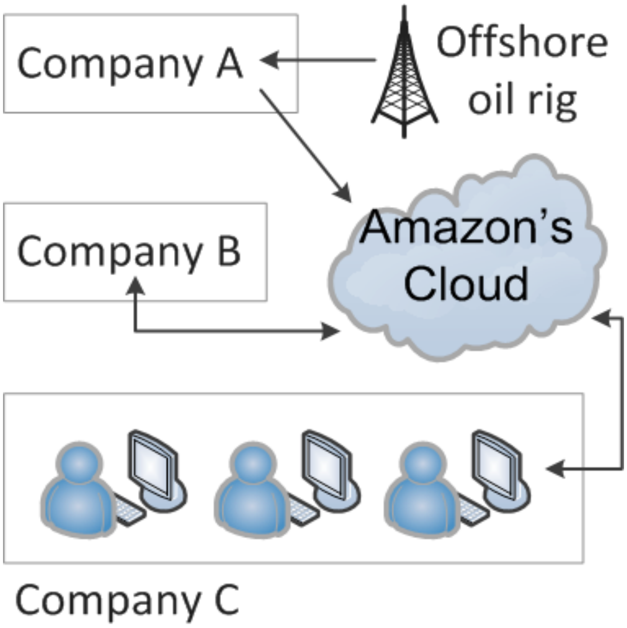
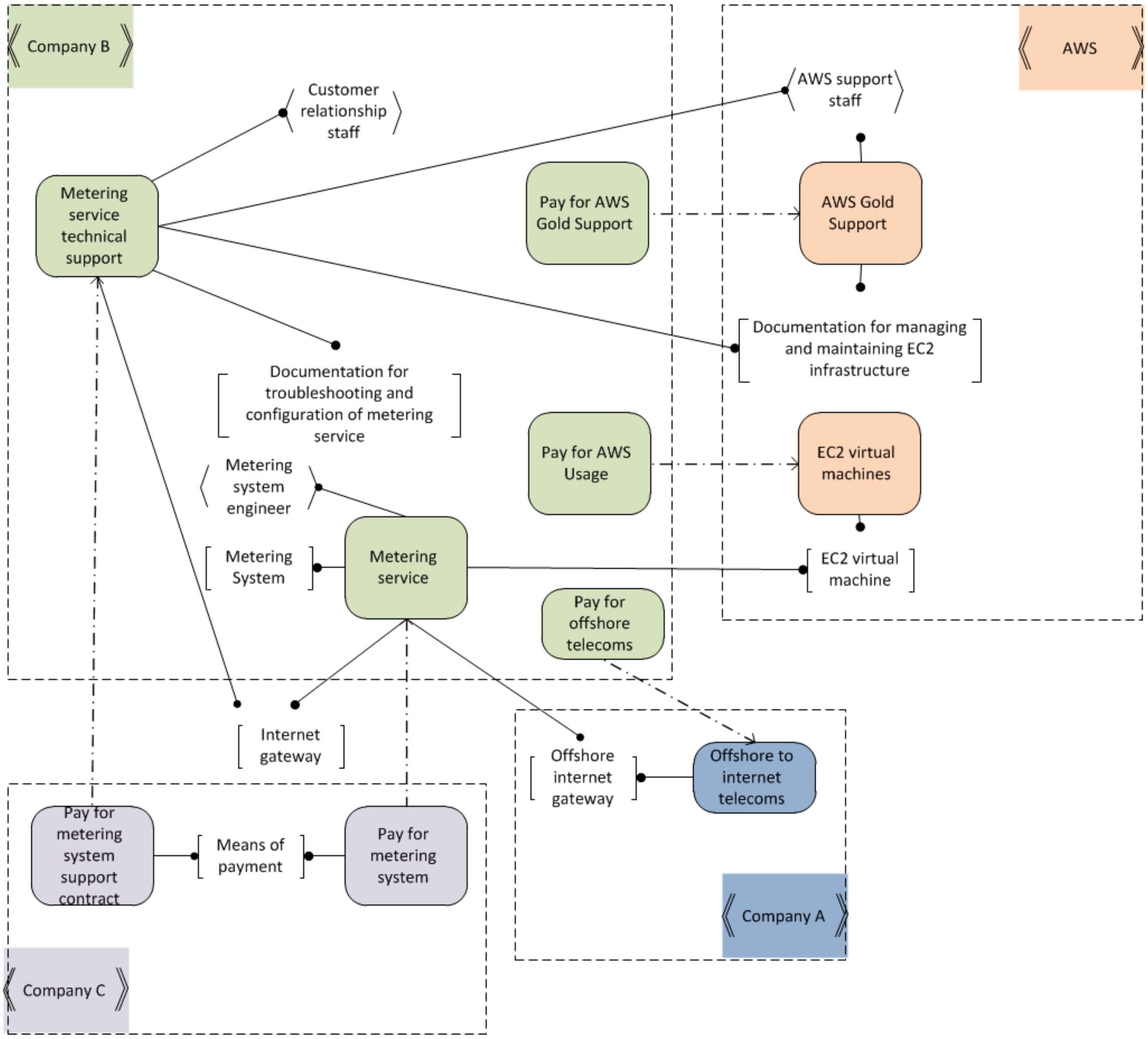


Figure 5.1 Logical structure of the 'to-be' system

5.4.2 Fieldwork and Results

To perform the responsibility-based threat identification we followed the two-part build-analyse process described in section 5.3. To do this we performed a series of interviews to identify the focal system of study, key stakeholders and their responsibilities. Our investigative remit was to consider the effect the deployment would have on Company B. Since we did not have permission to speak to suppliers or customers this limited our interviewees to: a project manager; a technical manager; a support manager; two members of support staff; and a business development manager. The interviewees were encouraged to discuss their concerns regarding the proposed project and also the opportunities that it could afford.

The resulting responsibility model (see Figure 5.2), which is a representation of the identified agents, resources and responsibilities (see Table 5.2 and Table 5.3), was then analysed to identify risk clauses (Tables 5.4 – 5.10) and interdependencies between risk clauses (Figure 5.3).



Company B
Customer relationship staff
Metering service technical support
Pay for AWS Gold Support
Documentation for troubleshooting and configuration of metering service
Metering system engineer
Metering System
Metering service
Pay for AWS Usage
Pay for offshore telecoms
Internet gateway
AWS
AWS support staff
AWS Gold Support
Documentation for managing and maintaining EC2 infrastructure
EC2 virtual machines
EC2 virtual machine
Offshore internet gateway
Offshore to internet telecoms
Company A
Pay for metering system support contract
Means of payment
Pay for metering system
Company C

Figure 5.2 - Responsibility Model of System

Figure 5.2 describes the configuration of responsibilities, agents and resources that composes the focal system. Organisational agents are represented by their name surrounded by double triangular brackets, responsibilities as rounded edged rectangles, resources by square brackets and human agents as single triangular brackets. The responsibilities that each organisational agent has a duty to perform are contained within a dashed rectangle. For clarity these organisational agents and responsibilities are also colour coded to remind the analyst of who is 'responsible for' for what. For instance we can see that 'Company B' is responsible for the metering service, metering service technical support, paying for AWS Gold Support, AWS usage and for offshore telecoms bills. We can also observe the resource dependencies of the responsibilities. For instance we can observe that to fulfil the responsibility of the metering service 'Company B' must have a metering system engineer e.g. an agent with the necessary skill set and availability to provision the service.

Table 5.2 describes the focal systems agents and their allocated responsibilities in terms of obligations, liabilities and norms. For instance one may observe that 'Company B' is responsible for attaining and maintaining the metering system and that 'Company C's responsibility to pay for the metering system depends on the 'Company A' fulfilling their responsibility. The relationship may also be observed graphically (Figure 5.2) as the *depends* relationship between the two responsibilities.

Table 5.2 - Specification of Agent Responsibilities

#	Agent	Obligation	Liability	Norm	Agent	Obligation	Liability
1	Company B	Attain and maintain **Metering system**	Financial liability for failing to meet SLA	within terms of SLA	Company C	Pay for metering system	Financial liability for breach of payment
2	Company B	Attain responses to **Metering system** requests for **technical support**	Financial liability for failing to meet SLA	within terms of SLA	Company C	Pay for metering system technical support	Financial liability for breach of payment
3	AWS	Attain and maintain **EC2 virtual machine**	If uptime based on "Regional Availability" < 99.5% then refund service credit amounting to 10% of monthly bill	"Regional Availability" excludes the failure of individual instances not attributable to "Region Unavailability".	Company B	Pay for AWS usage ~$0.085 per hour + other fees	Financial liability for breach of payment
4	AWS	Attain responses to **AWS Gold Support** requests	None. (No liability for advice nor non-adherence to support "guidelines")	Adhere to Premium Gold Support Guidelines -1 hour first response time for "urgent" issues	Company B	Pay for AWS Support $400 per month or 10% usage bill (greater amount)	Financial liability for breach of payment
5	Company A	Attain and maintain offshore to internet telecoms	Unknown	Unknown	Company B	Pay for offshore telecoms	Financial liability for breach of payment

Table 5.3 Specification of Agent and Resource Dependencies

#	Agent	Obligation	Agent / Resources	Agent	Obligation	Resources
1	Company B	Attain and maintain **Metering system**	-EC2 virtual machines -Internet gateway -Offshore internet gateway -Metering system -Metering system engineer	Company C	Pay for metering system	-Means of payment
2	Company	Attain responses to	-Metering system support staff	Company	Pay for metering	-Means of payment

	B	**Metering system** requests for **technical support**	-AWS support staff -Internet gateway -Documentation for troubleshooting and configuration of metering service -Documentation for managing and maintaining EC2 infrastructure	C	system	
3	AWS	Attain and maintain **EC2 virtual machine**	-EC2 virtual machines	Company B	Pay for AWS usage ~$0.085 per hour + other fees	-Credit card
4	AWS	Attain responses to **AWS Gold Support** requests	-AWS support staff -Documentation for EC2 infrastructure and configurations	Company B	$400 per month or 10% usage bill (greater amount)	-Credit card
5	Company A	Attain and maintain offshore to internet telecoms	-Offshore internet gateway	Company B	Pay for offshore telecoms	-Means of payment

Table 5.3 describes the resource dependencies of the responsibilities. For instance to fulfil the responsibility to attain and maintain the metering system the agent must have EC2 virtual machines, the internet gateway, the offshore internet gateway, the metering system and a metering system engineer. This is represented graphically (Figure 5.2) as the 'has relationship' between the responsibility, the resources and the agent.

Tables 5.4-5.10 described the 'risk clauses' identified as part of the analysis process. These were identified using HAZOPS keywords to encourage the analyst to consider the consequences of potential conditions on the fulfilment of each responsibility. Each table represents identified conditions associated with a specific resource or agent that could lead to a responsibility being reneged. For instance Table 5.4 describes the risk clauses for EC2 virtual machines and identifies 7 combinations of condition that may threaten AWS or Company B's responsibilities. An example risk clause from Table 5.4 consists of the an EC2 virtual machine being created late resulting in an availability threat which if not rectified results in Company B reneging on its responsibility to attain and maintain the metering system in accordance to its SLA.

Table 5.4 Risk clauses for EC2 Virtual Machine

#	Hazard Keyword	Condition	Threats	Severity	Recommended Action
1	**Late**	***EC2 virtual machine is late in its creation***	**CompanyB:MeteringService** **Availability Threat** -Customer temporarily does not receive service.	**Medium** **Company B Liability**: Possible financial liability if failing to meet SLA	When machine instance is available fetch back-log of polled data If server is significantly late then create an instance on temporary infrastructure
2	**Never / Incapable**	***EC2 virtual machine is never created***	**CompanyB:MeteringService** **Availability Threat** -Customer does not receive service. **AWS:EC2VirtualMachines** **Availability Threat** -AWS Region becomes unavailable	**Medium** **Company B Liability:** Possible financial liability if failing to meet SLA **AWS Liability:** If "region unavailable" for t > AWS SLA then service credit amounting to 10% of bill	Attempt to create another instance or create an instance on temporary infrastructure

3	**Insufficient**	***EC2 virtual machine has insufficient resources to execute application***	**CompanyB:MeteringService** **Reliability Threat** -Metering system would have degraded performance. **CompanyB:MeteringServiceTechnicalSupport Maintainability Threat** -Increased number of support calls.	**Medium** **Company B Liability:** Possible financial liability if failing to meet SLA	Attempt to create an EC2 instance with more resources. Transfer instance to in-house infrastructure.
4	**Impaired**	***Incorrect configuration of EC2 virtual machine is created.***	**CompanyB:MeteringService** **Reliability Threat** -Metering system may have degraded performance -Billing data may be come incorrect -Customer may receive degraded service **CompanyB:MeteringServiceTechnicalSupport Maintainability Threat** -Increased number of support calls.	**Medium** **Company B Liability:** Possible financial liability if failing to meet SLA	Create appropriate configuration of instance and remediate billing data prior to reaching customer account.
5	**Impaired**	***Incorrect EC2 infrastructure configuration***	**CompanyB:MeteringService** **Availability Threat + Reliability Threat+ Integrity Threat** -Metering system may have degraded performance -Metering system may become unavailable, unreliable or accessible by unauthorised parties **CompanyB:MeteringServiceTechnicalSupport Maintainability Threat** -Increased number of support calls. **AWS:EC2VirtualMachines** **Availability Threat** -Region may become unavailable **AWS:GoldSupport** **Maintainability Threat** -Increased number of support calls.	**High** **Company B Liability:** Financial liability for failing to meet SLA **AWS Liability:** If "region unavailable" then service credit amounting to 10% of monthly usage bill	
6	**Changes**	***EC2 services being used to support customers are withdrawn***	**CompanyB:MeteringService** **Accessibility Threat + Reliability Threat + Integrity Threat** -Customer may have service disrupted or degraded **CompanyB:MeteringServiceTechnicalSupport Maintainability Threat** -Increased number of support calls. **AWS:GoldSupport** **Maintainability Threat** -Increased number of support calls.	**Depends on changes** **Company B Liability:** Potential financial liability for failing to meet SLA or breach of contract. May be liable for cost of migrating to a different if necessary infrastructure.	Find alternative way of provisioning service to customers. Consider implementing back out plans to a different infrastructure.
7	**Changes**	***EC2 service offerings are changed but Company B not informed until after the change***	**CompanyB:MeteringServic Accessibility Threat + Reliability Threat + Integrity Threat** **-**Customer may have service disrupted or degraded **CompanyB:MeteringServiceTechnicalSupport Maintainability Threat** -Increased number of support calls. **AWS:GoldSupport** **Maintainability Threat** -Increased number of support calls.	**Depends on changes** **Company B Liability:** Potential financial liability for failing to meet SLA or breach of contract. May be liable for cost of migrating to a different if necessary infrastructure. **AWS Liability:** **None**	Contact EC2 on a regular basis to obtain information on changes to service offerings Consider implementing back out plans to a different infrastructure.

Table 5.5 - Risk clauses for Internet Gateway

#	Hazard Keyword	Condition	Threats	Severity	Recommended Action
8	Impaired	*Company B System engineers and support staff unable to access EC2 virtual machines*	**CompanyB:MeteringService** **Accessibility Threat + Reliability Threat** -Customer may have service disrupted or degraded as metering system engineers may not be able to access systems temporarily. **CompanyB:MeteringServiceTechnicalSupport** **Maintainability Threat** -Resolution of customer support calls is untimely	**Medium** **Company B Liability:** Possible financial liability if failing to meet SLA. **Gateway provider:** Liable for refunding monthly connection costs if SLA not met	Review SLA agreements with gateway provider and consider implementing fall-back gateway to provide redundant infrastructure

Table 5.6 – Risk clauses for AWS Gold Support Contract

#	Hazard Keyword	Condition	Threats	Severity	Recommended Action
9	Late	*EC2 Support staff respond with fix after the time period required*	**CompanyB:MeteringServiceTechnicalSupport** **Maintainability Threat** -Resolution of customer support calls is untimely	**High** **Company B Liability:** Financial liability for failing to meet SLA. May be liable for cost of migrating to a different if necessary infrastructure. **AWS Liability: None.** No liability for non-adherence to support "guidelines" nor advice given	Review SLA agreements with service provider and consider implementing back-out plan to alternative infrastructure
10	Never / Incapable	*EC2 Support staff does not respond with fix.*	**CompanyB:MeteringServiceTechnicalSupport** **Maintainability Threat** -Resolution of customer support calls is untimely -Support staff or system engineers must chase Amazon rather than performing fixes.	**High** **Company B Liability:** Financial liability for failing to meet SLA. May be liable for cost of migrating to a different if necessary infrastructure.	Review SLA agreements with service provider and consider implementing back-out plan to alternative infrastructure

Table 5.7 – Risk Clauses for Documentation for troubleshooting and configuration of metering service

#	Hazard Keyword	Condition	Threats	Severity	Recommended Action
11	Insufficient	*Documentation does not provide sufficient or adequate knowledge of Metering system to maintain*	**CompanyB:MeteringServiceTechnicalSupport** **Maintainability Threat** -Data acquisition system is not maintainable in the long term.	**High** **Company B Liability:** Financial liability for failing to meet SLA. Will need to invest in re-documenting the system or re-engineer the system so it may be documented.	Assess adequacy of documentation prior to launching the metering system in a production environment risk. Monitor fitness of purpose for system documentation.
12	Insufficient	*Available documentation does not provide*	**CompanyB:MeteringServiceTechnicalSupport** **Maintainability Threat** -Data acquisition system is not maintainable on EC2.	**High** **Company B Liability:** Financial liability for	Assess adequacy of documentation prior to migration and perform

		sufficient or adequate knowledge of EC2 infrastructure to maintain a commercial systems	**AWS:GoldSupport** **Maintainability Threat** -Support staff unable to assist users **AWS:EC2VirtualMachines Availability Threat** -Regions may become unavailable due to maintainability issues resulting in loss of availability	failing to meet SLA. May be liable for cost of migrating to a different service if necessary infrastructure. **AWS Liability:** None	pilots to minimise risk.
13	**Late**	***Available documentation does not provide sufficient or adequate knowledge of EC2 infrastructure to maintain a commercial systems***	**CompanyB:MeteringServiceTechnicalSupport Maintainability Threat** -Timely resolution of support calls is not manageable. **AWS:GoldSupport** **Maintainability Threat** -Timely resolution of support calls is not manageable.	**High** **Company B Liability:** Financial liability if failing to meet SLA. May be liable for cost of migrating to a different if necessary infrastructure**.** **AWS Liability:** None	-Contact Amazon for up-to-date documentation -Implement a back out strategy to switch to different infrastructure

Table 5.8 – Risk clause for Offshore Internet gateway

#	Hazard Keyword	Condition	Threats	Severity	Recommended Action
14	**Impaired / Insufficient**	***Offshore gateway becomes impaired such that valid data does not transfer***	**CompanyB:MeteringService** **Availability Threat + Reliability Threat** -Customer may have service temporally disrupted or degraded until causes can be identified and treated.	**Medium** **Company B Liability:** Financial liability if failing to meet SLA. May be liable for cost of migrating to a new gateway infrastructure if necessary. **Company A liability:** **unknown**	Assess adequacy of SLA with Company A. Consider whether risks are appropriately shared between parties.

Table 5.9 – Risk clause for Company C – Means of payment

#	Hazard Keyword	Condition	Threats	Severity	Recommended Action
15	**Insufficient**	**Company C Insufficient funds to pay for service**	**CompanyC:PayForMeteringSystem** -Payment not made **CompanyC:PayForMeteringSystemTechnical Support** -Payment not made	**Low** **Company C Liability:** Financial liability for missing payment.	NA

Table 5.10 – Risk clause for Company B – Means of payment

#	Hazard Keyword	Condition	Threats	Severity	Recommended Action
16	**Insufficient**	**Company B Insufficient funds to pay for service**	**CompanyB:PayForAWSUsuage** -Payment not made **CompanyB:PayForAWSGoldSuppor** -Payment not made **If payment late > 15 days CompanyB:MeteringService Accessibility Threat + Reliability Threat**	**Low** **Company C Liability:** Financial liability for missing payment. **High** **Company B Liability:**	Make payment as soon as possible

			-Customer may have service halted until payment made	Financial liability if failing to meet SLA.	

We identified 16 unique conditions that threatened the dependability properties (availability, reliability, integrity, maintainability) of the CoS. These included:

- Threats to availability due to the delayed creation of virtual machine instances, failures to create instances, misconfiguration of infrastructure, withdrawal of services, or notification of service withdrawal being delayed.

- Threats to reliability due to degradation of virtual machine instances, incorrect configuration of virtual machine instances, incorrect configuration of EC2 infrastructure, withdrawal of services, late notification of withdrawal of services.

- Threats to integrity due to misconfiguration of EC2 infrastructure, withdrawal of services, late notification of withdrawal of services.

- Threats to maintainability due to insufficient/inappropriate documentation, out-of-date documentation, degradation of virtual machine instances causing a surge of support calls, misconfiguration of virtual machine instances causing a surge of support calls, a misconfiguration of EC2 infrastructure preventing support engineers from connecting, 'Gold' support staff resolving issues in an untimely manner, and 'Gold' support staff incapable of providing a fix.

Of these threat conditions, four were present in both the 'as-is' and 'to-be' systems. These persistent threats comprise those due to: [11] insufficient or inadequate documentation to troubleshoot and configure the metering system; [14] offshore gateway becoming impaired such that valid data is not transferred; [15] company C has insufficient funds to pay for service; [16] company B has insufficient funds to pay for service. The other twelve threat conditions were unique to the EC2 configuration. For instance, [2] the EC2 virtual machine is late in its creation; [5] incorrect configuration of EC2 infrastructure; [6] EC2 services being used by customers is withdrawn.

To understand the possible interactions between the different identified risk clauses a causal map was used to represent how each risk contributes to another (if at all) – see Figure 5.3 below.

The map was created by first populating the model's nodes with the 'condition', 'threat' and 'severity' of each risk clause identified in tables 5.4-5.10. The links between the nodes represent that notion of 'possible cause' such that one node is a possible cause of another. To illustrate, using 'risk clause 1', we can see that node 1, links to node 17, which links to node 18. This represents 'risk clause 1', 'EC2 virtual machine is late in its creation', is a possible cause of 'Customer temporarily does not receive service', which is a possible cause of 'Company B Liability: Possible financial liability if failing to meet SLA'. By representing each of the identified risk clauses in this manner it enables the analyst to understand the interactions between risk clauses and also spot additional interactions may have not been apparent before.

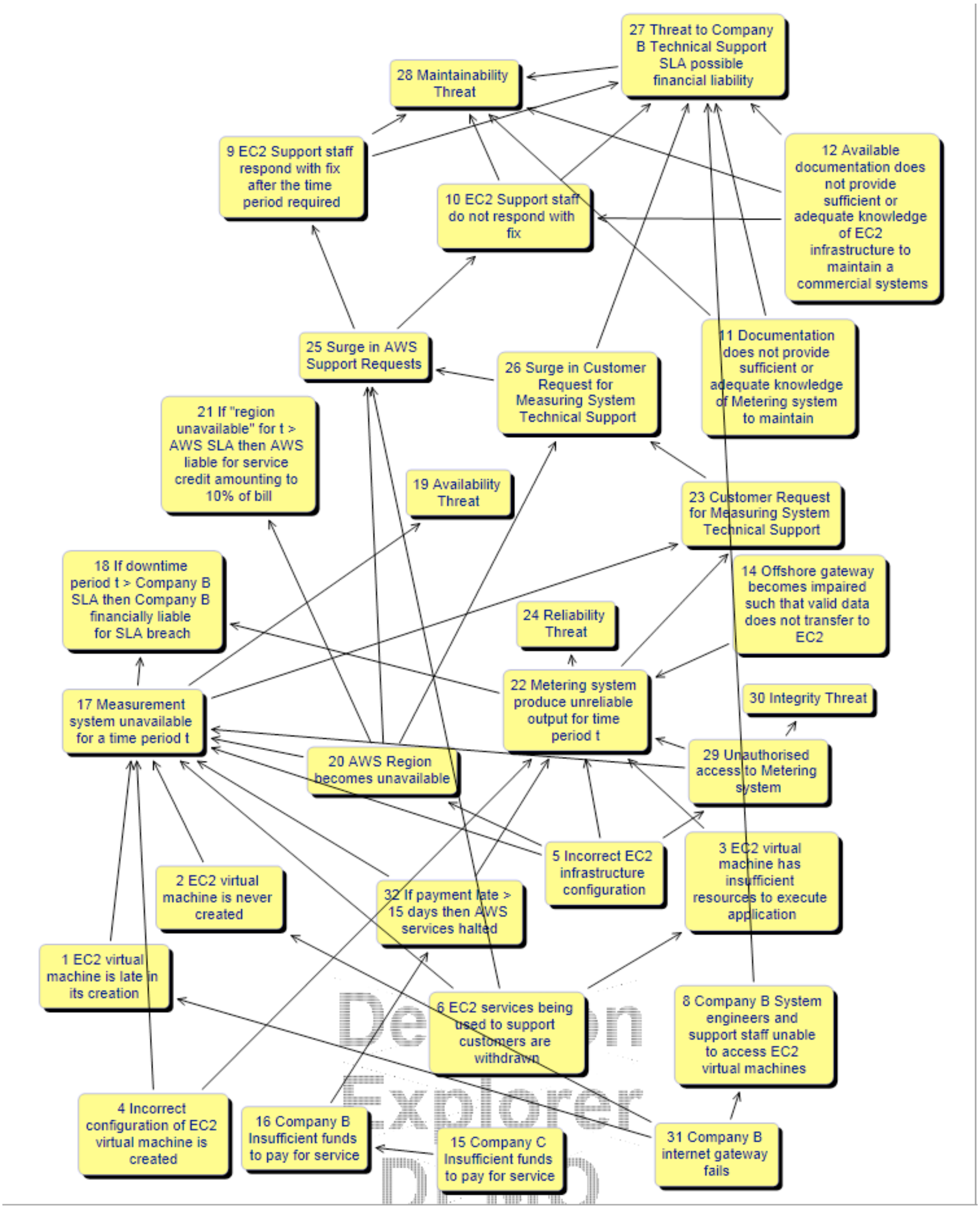

27 Threat to Company B Technical Support SLA possible financial liability
28 Maintainability Threat
9 EC2 Support staff respond with fix after the time period required
10 EC2 Support staff do not respond with fix
12 Available documentation does not provide sufficient or adequate knowledge of EC2 infrastructure to maintain a commercial systems
25 Surge in AWS Support Requests
26 Surge in Customer Request for Measuring System Technical Support
11 Documentation does not provide sufficient or adequate knowledge of Metering system to maintain
21 If "region unavailable" for t > AWS SLA then AWS liable for service credit amounting to 10% of bill
19 Availability Threat
23 Customer Request for Measuring System Technical Support
18 If downtime period t > Company B SLA then Company B financially liable for SLA breach
14 Offshore gateway becomes impaired such that valid data does not transfer to EC2
24 Reliability Threat
22 Metering system produce unreliable output for time period t
30 Integrity Threat
17 Measurement system unavailable for a time period t
20 AWS Region becomes unavailable
29 Unauthorised access to Metering system
5 Incorrect EC2 infrastructure configuration
3 EC2 virtual machine has insufficient resources to execute application
2 EC2 virtual machine is never created
32 If payment late > 15 days then AWS services halted
1 EC2 virtual machine is late in its creation
6 EC2 services being used to support customers are withdrawn
8 Company B System engineers and support staff unable to access EC2 virtual machines
4 Incorrect configuration of EC2 virtual machine is created
16 Company B Insufficient funds to pay for service
15 Company C Insufficient funds to pay for service
31 Company B internet gateway fails

Figure 5.3 Causal map of risk clause interactions

This form of representation is a particularly useful method of representing the behaviour of a system. This is because it enables a system analyst to explore the possible causes of undesirable behaviour e.g. it enables troubleshooting. Conversely it enables the exploration of the consequences of a component failing by ‘following the arrows forward’ e.g. risk analysis. This form of representation is also useful as it may be used to identify cascade failures, or feedback loops, that may result in unexpected emergent behaviour.

Since this system is a coalition-of-systems we also examined the fragility of the coalition partners overlapping self-interests. This was necessary as the on-going operation of the system, and so its dependability, depended on the organisational agents’ interests being satisfied by participation in the CoS. For this system since all the organisational agents’ are commercial organisations we assumed that their interest in the CoS was to make a profit that reflected the level of risk that they perceive to be taking i.e. the financial reward they expect outweighs the financial risk they are accepting by partaking in the CoS.

In order to assess whether the agents’ interests were being satisfied we inspected the system’s distribution of obligations and liabilities as described in Table 5.2. We noted that AWS provided little liability or warranty for their service offerings leaving the burden of liability with Company B. For instance, if AWS incorrectly configures their infrastructure resulting in degradation, or failure of individual machines, they take no liability for this even though Company B would be subject to financial liabilities due its SLA agreement with Company C. We noted even in situations where AWS accepts financial liability they will only refund 10% of monthly usage charges. This means that Company B is exposing itself to financial loss due to AWS failures unless their financial liability is less than 10% of their monthly usage charges.

This assessment of the distribution of liabilities indicates that the coalition’s overlapping self-interests may be highly fragile in the long term as AWS could renege on its responsibilities and suffer little financial loss whilst Company B would be left vulnerable. Company B could of course remedy this situation by requesting a stronger SLA from AWS such that a large proportion of the financial liability may be passed on to AWS; or by using multiple independent cloud partners to introduce an element of redundancy that reduces both the likelihood and severity of a cloud instance failure. In reality AWS is unlikely to offer a stronger SLA to Company B since it offers a commoditised service. However it is possible that a smaller cloud provider aiming to serve SMEs may offer such an SLA.

The lessons learned from this case study with respect to responsibility modelling were that:

1. Responsibility based threat identification may be used to identify conditions that threaten the required behavioural properties of SoS.
2. Responsibility based threat identification may be used to identify conditions that trigger cascades that threaten the required behavioural properties of SoS.
3. Responsibility based threat identification may be used to trace effect of changes to a systems behavioural properties to specific agents’ responsibilities.

The lessons learned from this case study with respect to the deployment of cloud-based systems were that:

1. Organisations assessing the feasibility of cloud solutions need to understand a cloud provider's obligations, liabilities and norms so that they understand what the provider responsible for and to what extent they are liable for reneging on these obligations.
2. Organisations should satisfy themselves that they are comfortable with the distribution of liabilities within a system. The distribution liabilities should be carefully examined so that organisations are aware of the others' self-interests and so can decide to take steps to preserve their own if necessary.

This case study highlighted a need to further extend agent responsibility-based threat analysis to improve its ability to identify and analyse agent related threat conditions. This case study primarily analysed *resource* related threat conditions, *agent* related threat conditions such as insufficient interest, insufficient skills, or insufficient time were not analysed in-depth. Organisational agents' interests were analysed by means of inspecting the distribution of liabilities, however individual agents' interests were not taken into account nor was the adequacy of their skills and their availability.

In the next chapter this weakness will be addressed using conflict analysis. Conflict analysis helps an analyst identify and understand potential conflicts of interest between human agents that may pose a threat to the behavioural properties of a system. These interests include a person's responsibilities, satisfaction and notions of fairness.

5.5 Conclusion

This case study of a cloud computing based CoS demonstrates that responsibility modelling when coupled with HAZOP keywords provides a means of identifying dependability threats associated with coalition partners reneging their responsibilities. This case study also demonstrates that causal maps may be used to identify potential cascade failures, or feedback loops, that could result in unexpected emergent behaviour or behavioural properties. It was also shown that inspecting the distribution of liabilities among coalition partners can indicate the fragility of overlapping self-interests.

6. Responsibility Modelling for Identifying Threats to Cooperation

6.1 Introduction

Responsibility models represent systems as a set of interacting agents that fulfil responsibilities with the aid of resources. For responsibilities to be fulfilled, agents may depend on the cooperative behaviour of other agents e.g. one agent provisioning a resource to another. Since a system's behaviour, or behavioural properties, may be dependent upon agents' fulfilling certain responsibilities, it is important to be able to identify threats that enable or impede cooperative interactions. One well-established method of understanding cooperation is through the lens of conflict theory (M. Deutsch, 1949; Morton Deutsch, 1973, 1990; Morton Deutsch & Coleman, 2006). In this chapter we use conflict theory to identify threats to cooperation that may arise because of conflicts.

This chapter's key contribution is to propose a responsibility based threat identification approach sensitive to different types of conflict. This builds on the work of the previous chapter, which focused on resource related threats, by providing a means of identifying agent related threats resulting from: agents holding incompatible responsibilities; being dissatisfied with their responsibilities; or perceiving that they are being treated unfairly.

A conflict-based framework is developed to identify these agent-related threats and two case studies are presented to illustrate its use. The first case study is a post-mortem of a well-known control room automation project (London Ambulance Service Computer Aided Dispatch) that was a public failure and subsequently turned into a success. This case supports the claim that the framework aids in the identification of threats due to incompatible interests and enables the identification of their consequences on system behaviour. The second case study the feasibility of a cloud adoption project at an oil/gas sector SME (from chapter 5) demonstrates the use of the technique to analyse a small yet non-trivial system-of-systems deployment. Overall both of these case studies support the claim that conducting a conflict analysis using the framework may provide insights that bound a system-of-systems potential behaviour.

A case study approach was selected because it illustrates that conflict analysis can be used in two non-trivial situations and thus highlights its practicality and usefulness to both academics and practitioners. Both case studies were selected to illustrate the use of the approach in different environments. The LASCAD case study is used to test the claim that the kinds of threats identified by conflict analysis if left unmitigated can have a significant effect on system behaviour. Our hypothesis was that the majority of conflict risks identified in the *failed*

LASCAD92 project would be unmitigated, whilst the majority of conflict risks would be appropriately mitigated/partially mitigated in the *successful* LASCAD96 project. Our results provided confirmation of this hypothesis indicating that the kinds of risk identified by conflict analysis are significant to the behaviour of systems composed of agents with differing interests. The oil/gas sector case study shows conflict analysis being used in a system-of-systems situation. The case study indicates that for little effort the approach is able to indicate unanticipated sources of potential conflict. This leads to the overall conclusion that conflict analysis does identify risks that are significant to the behaviour of systems composed of multiple agents with differing interests and may be used in differing contexts.

This chapter is structured such that section 2 introduces the reader to existing approaches and their limitations with respect to eliciting threats resulting from multiple agents holding differing interests. Section 3 presents the development of this thesis' approach to conflict analysis. Section 4 introduces our responsibility-based approach to identifying threats using conflict analysis. Section 5 presents the case studies illustrating that the approach identifies risks that are significant to the behaviour of systems composed of multiple agents with differing interests.

6.2 Background

Existing approaches to socio-technical threat identification do not identify threats related to a system being incompatible with agents' interests. The most notable socio-technical risk analysis approaches relevant to analysing socio-technical situations are Functional Resonance Analysis Method (FRAM) (Hollnagel, 2004; Hollnagel & Goteman, 2004), Systems-Theoretic Accident Modelling and Processes (STAMP) (Leveson, 2004; Leveson et al., 2006) and Responsibility Modelling for Risk Analysis (Lock et al., 2009; Sommerville et al., 2009).

The FRAM approach is a process level accident analysis and risk analysis method. It uses the concept of interacting 'functions' to represent a process and identify risks to the outcome of a process. FRAM is performed by dividing a process into a number of interacting functions comprising inputs, outputs, preconditions, control constraints, timing constraints and resources. The potential variability of functions is identified and their implications are noted as consequences on the outcome. FRAM has been used to identify risks and analyse accidents involving socio-technical systems including air accidents and medical accidents (Hollnagel et al., 2008).

The STAMP approach is an institutional level accident analysis and risk analysis method. It uses the concept of interacting parts in dynamic equilibrium to represent institutional structures and identify risks in terms of 'control problems', the premise being that risks arise when a system's behaviour is not appropriately monitored and controlled. Identified risks are analysed in terms of their interactions with the systems control structures and their resultant effect on institutional outcomes. STAMP has been used to analyse high profile socio-technical accidents including the loss of Space Shuttle Columbia and the Walkerton water contamination tragedy (Leveson, Daouk, Dulac, & Marais, 2003).

The responsibility modelling approach is a tactical level risk analysis method. It uses the concept of responsible agents (human / organisational agents) and their interactions to represent a situation and identify risks in terms of failures of agents to fulfil responsibilities. Risk identification is performed by: 1) modelling the responsibilities of the agents involved in a situation and the resources they require to discharge their responsibilities; 2) identifying the consequences/liabilities resulting from of an agent not having a resource or not discharging a responsibility. This can be facilitated by the use of hazard keywords such as early, late, never, incapable, insufficient and impaired. Responsibility modelling has been used to analyse socio-technical systems including E-counting systems in the Scottish elections and UK civil emergency plans [14, 15].

Responsibility modelling is a technique that is complementary to both STAMP and FRAM. STAMP may be used to identify risks at an institutional governance level whilst FRAM may be used to identify risks at a process level. Responsibility modelling enables the analysis of situations at a ‘tactical level’ by understanding the risks related to agents depending upon others to discharge their responsibilities.

Responsibility modelling offers a number of attractive characteristics that makes it more suitable for conflict analysis of SoS than either FRAM or STAMP. Firstly, developing, deploying and maintaining a SoS necessarily requires reliance on other parties to discharge responsibilities. Secondly, responsibilities are relatively unproblematic to elicit as people find them ‘natural’ to articulate in comparison to ‘technical’ constructs such as functions or goals. Thirdly, responsibility modelling is relatively rapid to perform unlike FRAM. FRAM is concerned with process level risks and therefore elicits information such as functions, preconditions, control constraints and so on, which may be impractical for large systems. Fourthly, STAMP is unsuitable as it focuses on institutional level control structures to identify ‘control problems’. This assumes that there already exist appropriate techniques to identify conflict-related threats so that they can be controlled.

6.3 A Framework for Identifying Conflict-related Threats

Responsibility models represent systems as a set of interacting agents that fulfil responsibilities with the aid of resources. For responsibilities to be fulfilled, agents may depend on the cooperative behaviour of other agents e.g. one agent provisioning a resource to another. Since a system’s behaviour, or behavioural properties, may be dependent upon agents’ fulfilling certain responsibilities, it is important to be able to identify factors that enable or impede cooperative interactions as part of threat identification. One well-established method of understanding cooperation is through the lens of conflict theory (M. Deutsch, 1949; Morton Deutsch, 1973, 1990; Morton Deutsch & Coleman, 2006). In this section we use conflict theory to identify threats to cooperation that may arise because of conflicts. Thus according to this method of analysis:

> A system's desired behaviour, or behavioural property, is said to be threatened if it is judged that it is '*unlikely for all necessary agent interaction to be successful due to the presence of threats that promote conflict or impede its resolution*'.

To identify factors that are important to the occurrence of conflict, a literature survey was performed by searching for the term conflict and/or resistance in the following journal databases: MISQ, ISJ, Journal of Personality & Social Psychology, Journal of Applied Psychology, Journal of Management, Administrative Science Quarterly, Academy of Management Journal, Information Management, CSCW, CACM, Behaviour & Information Technology, Management Science, Information Technology and Management. The rationale for approaching conflict analysis from this multidisciplinary perspective was for rigorously designed studies from experimental psychology, social psychology, and insights from administrative science and management, to be brought to bear upon the problem and therefore provide a sound theoretical basis for analysis.

This section is structured so that a summary of the survey findings is presented up-front to orient the reader. Following this orientation detailed discussion follows. This detailed discussion covers the structure of cooperative and conflictual situations, types of agent interest and behaviours that affect these interests. It turns out that understanding these factors is important to the analysis of conflict.

6.3.1 Summary of Literature Review Findings

Techniques for socio-technical threat identification should take into account the following insights from conflict research to enable the identification of conflict related threats:

1. The extent that a system is exposed to conflict is dependent on the system's agents' interactions' *potential for conflict* and *potential for a cooperative resolution*.

2. An interaction's *potential for conflict* is a function of:

- The number and importance of the perceived sources of conflict
- The following are identified sources of agent conflict
    - Task conflict, process conflict, time, capabilities/skills, resources, values, satisfaction
    - procedural / distributive injustice

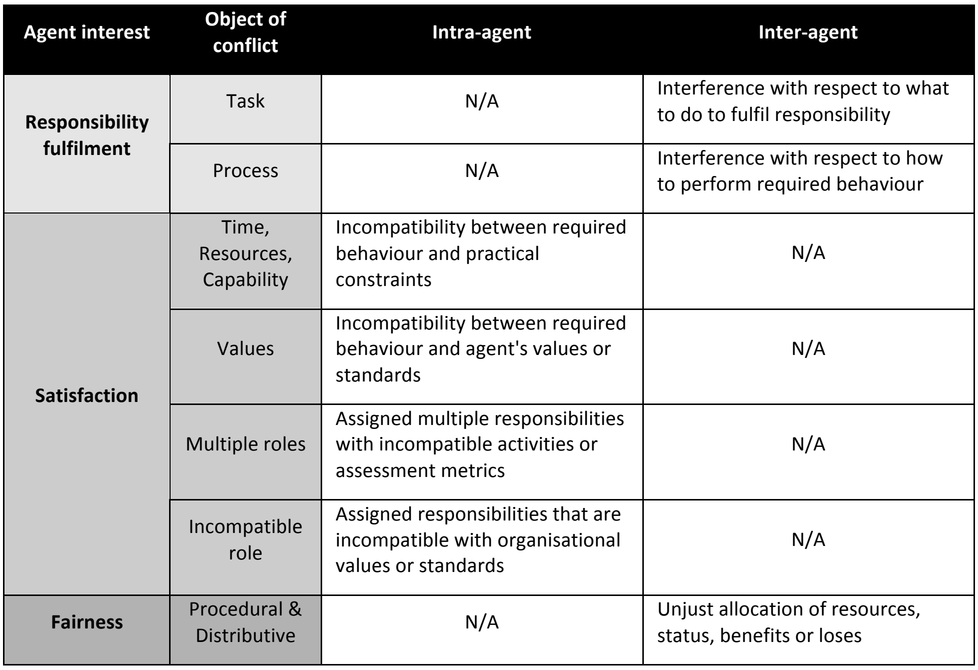
3. An interaction’s *potential for a cooperative resolution* is dependent upon these following factors: outcome interdependence, importance, acceptability of conflict, power, temporality, organisational barriers and history of conflict.

- A cooperative resolution is dependent upon positive outcome dependence between stakeholders.
- Intensity of cooperative or competitive conflict is dependent upon task importance.
- Stakeholders in equal positions of power are more likely to be exploitative than those in power asymmetries.
- A positive outcome from conflict is dependent being neither at the start of end of a project
- Conflict resolution is dependent upon good quality communication as poor quality communication can be perceived as political manoeuvring. Perceived non-cooperative behaviour can occur because of theme incompatibilities, language differences, incomplete or specialised knowledge.

4. Effect of conflict on systems:

- Any type of agent conflict is likely to result in disruption of work as conflict has negative effects on work activity and individual satisfaction with exception to decision quality in certain cases.

Table 6.1 - Summary of factors that affect potential for conflict



Agent interest	Object of conflict	Intra-agent	Inter-agent
Responsibility fulfilment	Task	N/A	Interference with respect to what to do to fulfil responsibility
	Process	N/A	Interference with respect to how to perform required behaviour
Satisfaction	Time, Resources, Capability	Incompatibility between required behaviour and practical constraints	N/A
	Values	Incompatibility between required behaviour and agent's values or standards	N/A
	Multiple roles	Assigned multiple responsibilities with incompatible activities or assessment metrics	N/A
	Incompatible role	Assigned responsibilities that are incompatible with organisational values or standards	N/A
Fairness	Procedural & Distributive	N/A	Unjust allocation of resources, status, benefits or loses

6.3.2 The Structure of a Cooperative Situation

One way of understanding the structure of a cooperative situation is to use (M. Deutsch, 1949) theory of conflict. According to (Morton Deutsch, 1973), conflict comprises situations where one person’s actions interfere, obstruct, or in some way get in the way of another’s action. (M. Deutsch, 1949) theorised and then later demonstrated that the extent that people perceive their goals as interrelated is a good predictor of the consequences of interaction. Cooperation occurs when agents perceive their *interests* as positively dependent and that their *actions* do not interfere with each other. Cooperative conflict occurs in situations where agents perceive their interests as positively interdependent but their actions interfere. Competition occurs in situations, where the converse is true, people perceive their interests to be negatively dependent but their actions compatible. Competitive conflict is said to occur in situations where agents perceive that their interests are negatively dependent and their actions interfere.

Table 6.2 - The structure of a cooperative situation

	Positive Goal Interdependency	Negative Goal Interdependency
Action Interference / Obstruction	Cooperative Conflict	Competitive Conflict
Action Compatibility	Cooperation	Competition

In summary, the extent that agents composing a system will conflict is dependent upon the interdependency of their interests and the extent that their actions interfere with one another's behaviour. Therefore, to understand the threats to the behaviour of a system resulting from agents' differing interests we need to develop a framework to guide the identification of the:

1. types of interest that are sufficiently important to agents that they will defend them using conflict;
2. types of behaviour that are typically perceived to interfere with an agents interests.

6.3.3 Determinants of Potential Conflict

Studies of conflict suggest that there are three broad types of interest that if interfered with may result in conflict. These interests are responsibility fulfilment, satisfaction and fairness.

6.3.3.1 Responsibility Fulfilment

Responsibility fulfilment is often associated with task and process conflict. Studies of task and process conflict suggest that people conflict when their ability to fulfil a responsibility is interfered with. The kinds of actions that interfere with this interest include:

1. disagreements over *what* should be done to fulfil a responsibility;
2. disagreements over *how* it should be done;
3. disagreements over *who* should do it;

4. disagreements over what *resources* are required.

Task conflict is one of the most commonly identified types of conflict (Janssen, Vliert, & Veenstra, 1999; Jehn, 1995, 1997; Jehn & Mannix, 2001; Moeller & Zhang, 2008; Tjosvold, Poon, & Yu, 2005). There is much agreement about its primary focus but researchers draw boundaries around the concept in different ways. For example (Janssen et al., 1999, p. 119) states that "Task conflict in team decision making refers to disagreements about the work to be done including issues such as the allocation of resources, application of procedures, and the development and implementation of policies." Whilst (Jehn & Mannix, 2001, p. 238) states that "Task conflict is an awareness of differences in the viewpoints and opinions pertaining to a group task. … it pertains to conflict about ideas and differences of opinion about the task". Notice that Janssen includes conflict about what work is to be done but also allocation of resources and to how to do it. In contrast, Jehn limits the concept to ideas and opinions about the task. This thesis promotes the usage of a common language to ensure comparability of results and thus suggests that task conflict should be regarded as conflict over what to do (e.g. to fulfil a responsibility) but should not include how-to do it which is the object of conflict during process conflict.

The term process conflict (Greer & Jehn, 2007; Jehn, 1997; Jehn & Mannix, 2001) is often used to add granularity to the study of conflict. It is concerned with conflict surrounding responsibility accomplishment e.g. a group may be assigned the responsibility of reducing operating costs by 10% and therefore process conflict could arise about how to do that. For example (Jehn, 1997, p. 540) describes it as "conflict about how task accomplishment should proceed in the work unit, who's responsible for what, and how things should be delegated". Process conflict includes disagreements about assignments of duties or resources (Jehn, 1997). This definition of process conflict, we believe, conflates two separate yet important issues. The first is developing a strategy for how to do the task. The second is implementing the strategy by assigning who is to perform certain roles in the task. Therefore, this thesis proposes that process conflict should be restricted to conflict over strategies for how a task should be performed and that conflicts over implementation (e.g. who and assignment of resources) should be dealt with as separate issues.

6.3.3.2 Satisfaction

Studies of role conflict (Rizzo, House, & Lirtzman, 1970) suggest that people conflict when their job satisfaction is inferred with. In this context a role is a set of responsibilities that a person or organisational agent must fulfil. Our review of role conflict identified that the following actions are triggers of role conflict:

1. a person being assigned a role that is incompatible with their internal standards or values;

2. a person being assigned a role that is incompatible with their available time, resources or capabilities;
3. a person being assigned a role consisting of multiple responsibilities which require incompatible behaviour;
4. a person being assigned a role that is incompatible with organisational policies, procedures and standards.

6.3.3.3 Fairness

Studies of procedural justice and distributive justice suggest that people also conflict when they perceive that they are being treated unfairly. Procedural justice refers to conflict over procedures that determine the relationships between parties. According to (Brown, 2000) it is "a feeling that the methods for deciding about and allocating material goods are unfair independently of the in-group's actual outcomes". This thesis proposes that procedural injustice is not limited to material goods but also social status, time and finite organisational resources such as money or manpower. A second type of injustice is distributive injustice where parties perceive the distribution of benefits/losses to be unfair such that they enter into conflict with the those perceived to control their distribution (Brown, 2000).

6.3.3.4 Empirical Studies of Conflict in Systems

Studies of systems development confirm that these types of conflict occur in system developments. (Butler, 2003) illustrates that conflicts between groups emerge and create a high degree of institutional tension when an IT system is used to implement organisational capabilities that are not perceived as being in the interest of certain groups. For example, Butler (2003) found that certain groups perceived the introduction of an e-commerce capability as an interference with tried and tested face-to-face methods of sales and thus a threat to their status within their organisation. (Ellingsen & Monteiro, 2003) provides a similar account from the deployment of a hospital patient record system that resulted in different clinical coding systems being used across different systems due to doctors and administrators having conflicting needs for coding. (Lawrence, 2006) supports this with an account of conflict over the role of a software development team within a large distributed e-Science project. The team were invited to join a project to develop software to support meteorologists. However, the team received their funding for this work from a research grant that interfered with their responsibility to develop software. The research grant was given to the group to pursue research but the group were allowed to join the project to perform software development work. This created a perceived incompatibility between the roles expected of them by the funders, the project's managers and collaborators.

Stakeholder agendas, identities and values have also been studied in detail during the development and testing phase of ISD (Cohen, Birkin, Garfield, & Webb, 2004; Sonnenwald, 1995; X. Zhang, Dhaliwal, Gillenson, & Moeller, 2008). The sources of conflict between developers and testers comprise primarily of differences of: identity leading to a perceived asymmetry of status; differences in perceptions/expectations of appropriate extent of testing; limited resources.

Stakeholder agendas, identities and values are also a source of conflict in the context of software evaluation and system deployment (Joshi & Rai, 2000; Wong, 2005). Users and developers despite wanting similar consequences from software (e.g. look good at job, less stress, flexible, accurate) were motivated by different values and believed that different software characteristics would deliver those desired consequences (Wong, 2005). Within system deployment it is known that software that attempts to make employees do activities which they perceive not to be part of their role, or incompatible with their values, causes role conflict and is negatively correlated with employee satisfaction (Joshi & Rai, 2000).

6.3.3.5 Summary of Determinants of Conflict

In summary, the types of interest that are sufficiently important to agents' that they will defend them using conflict are:

- fulfilling responsibilities
- job satisfaction
- fairness

The types of actions that are typically perceived to interfere with an agent's interests are:

- disagreement over *what* should be done to fulfil a responsibility;
- disagreement over *how* it should be done;
- disagreement over *who* should do it;
- disagreement over what *resources* are required;
- being assigned a role that is incompatible with their internal standards or values;
- being assigned a role that is incompatible with their available time, resources or capabilities;
- being assigned a role consisting of multiple responsibilities which require incompatible behaviour;
- being assigned a role that is incompatible with organisational policies, procedures and standards;
- perception of unfair *procedures* for distributing benefits or loses;
- perception of unfair *distribution* of benefits or loses.

Having identified determinants of potential for conflict in this section, the next section identifies the determinants of potential for a cooperative resolution of a conflict. Understanding cooperative resolution is important to conflict analysis because conflicts that are unlikely to be resolved pose a greater threat to a system than those that are likely to be resolved.

6.3.4 Determinants of Potential for Cooperative Resolution

This subsection provides an account of the factors that determine a conflict's potential for resolution. Our survey identified the following as determinants of the potential for resolution: structural factors; temporal factors; communicative factors; individual factors.

6.3.4.1 Structural Factors

Structural factors are factors that describe the relationship between the agents in a situation. In conflict situations outcome interdependence, task interdependence, task importance and power relations are important factors in determining the outcome of a conflict.

Our survey suggests that a cooperative resolution to a conflict is dependent upon positive outcome interdependence between stakeholders. In general, having a high level of positive interdependence between conflicting parties increases decision quality and affective acceptance of outcome especially in situations with high personal conflict (Janssen et al., 1999).

- Positive interdependence has been measured to mildly decrease decision quality and affective acceptance in situations with high task conflict and low personal conflict.
- Positive interdependence promotes integrative behaviour in situations with low task conflict (or high task conflict & personal conflict). This means positive interdependence promotes behaviour that maximises total outcome for parties involved.
- Positive interdependence promotes reductions in distributive behaviour in situations with low personal conflict (or high task & personal conflict). This means positive interdependence discourages behaviour that maximises unequal outcome for the respective parties.

The implication for risk analysis is that positively dependent stakeholders are more likely to reach an acceptable resolution than those negatively dependent.

The speed of a cooperative resolution is affected by importance of the outcome that is being interfered with or inhibited. The importance of the outcome amplifies the positive and negative effects of a conflict and may speed up resolution (Jehn, 1997). If an organisation's norms accept conflict then this also typically amplifies both the positive and negative impacts of conflict (Jehn, 1995, 1997).

The likelihood of a cooperative resolution is also affected by agents' relative power. Agents in equal positions of power are more likely to be exploitative than those in power asymmetries. The importance of power has been demonstrated experimentally (Solomon, 1960). The greater the social power of an agent in contrast to another, the more likely they are to engage in trusting behaviour. Under conditions of equal power an agent is likely to be exploitative if the other party unconditionally cooperates, whilst cooperative if other party cooperates on condition. In unequal power conditions, the opposite is true. If the other party cooperates unconditionally then so will the agent. The implication for risk analysis is that agents in equal positions of power are more likely to be exploitive and thus increase the likelihood that conflicts will remain unresolved.

The effects of structural factors have been observed and studied in the field of Computer Science and Information Systems. Studies of user participation within the development process support claims that structural factors mediate conflict and promote cooperative stakeholder behaviour in organisational settings (Daniel Robey, Farrow, & Franz, 1989).

(D. Robey, Welke, & Turk, 2001) argue that the traditional life cycle, iterative incremental and component based development all have qualitatively different approaches to user participation/communication and therefore the manner in which conflict emerges is different. The traditional life cycle paradigm uses a carefully designed social process that relies on a largely sequential process that suppresses conflict by creating a power-relationship between developer and user. The developer controls the sequential interaction with users thereby suppressing 'off-topic' issues that become rarely addressed.

The iterative incremental paradigm uses a structured process to bring together the developer and user. However, the interactions are less structured thereby giving the user more power to direct attention to conflict issues than in traditional development. It is also argued that users and developers are more interdependent as users assist design and therefore this provides shared goals that promote cooperative behaviour.

The component based development paradigm uses a structured process to bring together the developer and users with the aim of removing users dependence on in-house IT staff and thus giving users the power to build complete systems from components with little interference (D. Robey et al., 2001). This claim however is disputed by (Garlan, Allen, & Ockerbloom, 2009). Garlan *et al.* suggest that under the component based paradigm, conflicts are addressed through buyer seller negotiation, but due to end-users' lack of formal training to deal with companies, it results in them using intermediaries or them becoming dependent on suppliers or component brokers, thereby preventing users from building systems with minimal interference.

6.3.4.2 Temporal Factors

A cooperative resolution is influenced by the time period within a project that the conflict appears. Higher group performance is associated with a specific pattern of conflict (Jehn & Mannix, 2001). A high performing group at the beginning of a project will have low but increasing levels of task conflict, a low level of relationship conflict and a low level of process conflict. At the midpoint of the project there will be moderate levels of task conflict and this will subside towards the end of the project. These high performing teams also had established value systems, high levels of trust and respect, and open discussion norms. These findings are supported by the work of (Tjosvold et al., 2005) who identify that prior cooperative conflict (consensus based conflict resolution) promotes confidence in team relationships (as perceived by team members) and promotes team effectiveness (as perceived by team's manager). Conversely low group performance is associated with a particular pattern of conflict. It begins with low task, relationship, and process conflict and remains low until the end of the project where conflict peaks (Jehn & Mannix, 2001).

6.3.4.3 Communicative Factors

A cooperative resolution is dependent upon good quality communication as poor quality communication can be perceived as political manoeuvring (X. Zhang et al., 2008). Intra-group communication promotes cooperation in social dilemmas (Kerr & Kaufman-Gilliland, 1994). Experimental results support that communication is believed to increase cooperation via individuals making commitments and building trust not by the development of group identity (Kerr & Kaufman-Gilliland, 1994). When actors are left out of the loop, or receive last minute communication, or delayed communications this can be perceived as a provocation. Common examples of this include delays communicating changes to requirements, developers changing code without notifying testers, testers failing to provide feedback to developers, developers failing to communication with users (X. Zhang et al., 2008).

Conflict can also occur because of theme incompatibilities, language differences, and incomplete or specialised knowledge (Sonnenwald, 1995). Theme incompatibility occurs when one party is unable to answer questions in another's terms. For example, a user asks how much a particular feature will cost and the developer responds by explaining the standard pricing model of function points rather than the price in terms of cash. Language differences can cause parties to believe commitments or assurances have been made when they haven't resulting in conflict. Incomplete or specialised knowledge can be perceived as obfuscation when in fact a certain variable in unknown or uncertain/vague in nature. Conflicts can also occur because of blunt communication that is perceived as an attack upon an individual or group's status/reputation (Kock & McQueen, 1998).

6.3.4.4 Individual Factors

A cooperative resolution is also affected by a stakeholder's perceptual frame type as it influences the extent to which a person is likely to share the benefits (and losses) of interaction (De Dreu & McCusker, 1997). A perceptual frame is the 'frame of reference' that a person uses to analyse a situation. An actor is said to be in a gain frame where prospective outcome is gain e.g. gaining points. An actor is said to be in a loss frame where a prospective outcome is less favourable than reference e.g. losing points. De Dreu demonstrated that humans exhibit three kinds of behavioural strategy labelled: pro-social; individualist; competitive. Pro-socials in a gain frame are likely to share loses evenly but not gains. In a loss frame, pro-socials tend to cooperate more. Individualists in a gain frame are more likely to share gains evenly but are less likely to share losses. In a loss frame they cooperate less. Actors of a competitive nature cooperate as little in a loss frame as a gain frame.

6.3.4.5 Summary of Determinants of Cooperative Resolution

In summary, the potential for a cooperative resolution is dependent upon certain conflict dimensions that impact the conflict outcome and intensity. These dimensions include outcome interdependence, importance, acceptability of conflict, power, temporality, organisational barriers and history of conflict.

Table 6.3 - Summary of determinants of cooperative resolution

Factor Type	Determinant of Outcome / Intensity	Affect
Outcome interdependence	Negative interdependence between Stakeholders	Reduced likelihood of resolution
Importance	High importance of issue to Stakeholder	Conflict is highly disruptive to work
Acceptability Norms	Perception that acceptability of conflict is high	Conflict is highly disruptive to work
Power	Equal power between stakeholders	Reduced likelihood of resolution
Temporal	Conflict at the start or end of a project	Reduced likelihood of resolution
Organisational Barriers	Communication quality between parties is below average e.g. Specialist knowledge, different norms.	Reduced likelihood of resolution
History of conflict	History of dislike / unresolved conflict	Reduced likelihood of resolution

- A cooperative resolution is dependent upon positive outcome dependence between agents.
- Intensity of conflict is dependent upon outcome importance.

- Agents in equal positions of power are more likely to be exploitative than those in power asymmetries.
- A positive outcome from conflict is dependent on it being neither at the start or end of a project
- Conflict resolution is dependent upon good quality communication as poor quality communication can be perceived as political manoeuvring. Non-cooperative behaviour can occur because of theme incompatibilities, language differences, incomplete or specialised knowledge.

6.4 Responsibility Modelling for Conflict Analysis

Responsibility-based conflict analysis aims to identify conditions that result in a system not exhibiting required behaviours or behavioural properties due to agent conflict. It shares many similarities to responsibility modelling for threat identification described in the previous chapter. A system is represented as an aggregate of agents, resources and responsibilities. Required behaviours are represented by responsibilities which represent units of behaviour that an agent has a duty to perform; a responsibility model is thus a description of the behaviours that agents (within the scope of analysis) have a duty to perform and the resources they require to perform these behaviours.

Responsibility-based conflict analysis differs from the threat identification performed in previous chapters during data collection and analysis. During data collection information must be gathered to identify potential conflict and conditions that may inhibit their resolution. During data analysis this information is used to evidence claims about the extent that each agent's interests are likely to be interfered with and whether these conflicts are likely to be resolved in a cooperative manner.

6.4.1 Building a Responsibility Model

As per the previous chapter, the responsibility models may be developed and analysed in any manner that is flexible enough to meet an analysts needs.

6.4.2 Analysing a Responsibility Model for Conflict

Analysis of a responsibility model for threats may be performed as per the previous chapter. In addition to the steps described in the previous chapter this time responsibility fulfilment should also be inspected for *agent related threat conditions* resulting from *conflict*. This comprising collecting information to assess whether agents have sufficient time, sufficient/adequate skills, and that their interests are compatible with their assigned responsibilities.

Conflict analysis identifies threats to each agent's responsibility fulfilment interests, satisfaction interests, and fairness interests. This is achieved by judging whether: i) the agent's responsibilities are likely to be compatible with their internal standards and values; ii) whether the agent's responsibilities are compatible with their time, resources and capabilities; iii) whether the combination of responsibilities that have been assigned to an agent are compatible with each other; iv)whether the responsibilities assigned to an agent are compatible with organisational standards and policies; and v) whether the agent will perceive changes as unfair or reduces their satisfaction. This form of analysis is supported via means of conflict analysis checklist that is completed for each agent (See Appendix A).

6.5 The Case Study Investigations

Two case studies were performed to investigate the usefulness of conflict analysis. The first case study comprised a retrospective analysis of a well-known and well-studied control room automation project (LASCAD) that suffered from failure prior to being successfully turned around. The second case study comprised a feasibility study of a proposed migration from an in-house data-centre to a cloud-based infrastructure as a service (IaaS). The aim of the first case study is to verify the reliability of conflict analysis by mapping the findings of previous studies to the threats identified using conflict analysis, and to verify that the types of threats identified using conflict analysis impacted the behaviour of the system. The aim of the second case study is to demonstrate that the technique can be used to analyse a real world system-of-systems.

6.5.1 The Turnaround of the LASCAD Project

The LASCAD (London Ambulance Service Computer Aided Despatch) project is an exemplar of an IT enabled work-transformation project. It comprised the automation of the dispatch of ambulances from call taking to ambulance dispatch. It is a particularly interesting case study since the project was a failure in 1992 but was then subsequently turned around and made into a success in 1996. This case study was selected as the organisational environment of the project was particularly complicated (as shall be described below) and the project itself is relatively well known and well studied in information systems and computer science. This case study will be used to test the reliability and validity of the conflict analysis approach. If the approach provides valid and reliable output, one expects that conflict analysis will identify a large number of unmitigated threats that map onto those threats found by other analyses of the LASCAD92 project. In contrast, for the successful LASCAD96 project, one would expect that a large number of those threats would have been mitigated.

6.5.1.1 History of the LASCAD Project

The need for the automation project was identified in the mid-1980s when the government perceived the London Ambulance Service to be failing to modernise and generally invest in their work force. In May 1987, a project was initiated to automate ambulance dispatch from call taking to resource allocation. In October 1990, this attempt to automate was scrapped. IAL, a British Telecoms subsidiary, was blamed for faulty software. LAS sought damages for a faulty dispatch module that failed under load testing (P. Beynon-Davies, 1995). In October 1990, a second LASCAD project was initiated. By June 1991 a contract with IT developers, Systems Options Ltd, was signed and by September 1991 a contract with a mobile data equipment provider, Solo Electronic Systems Ltd, was also signed. The planned implementation date was 8th Jan 1992 but by March 1992, the second phase of live trials was suspended due to the users of the system not having confidence in the system, resulting in the Nation Union of Public Employees to become involved (Page, Williams, & Boyd, 1993). On the 26th October 1992, the LASCAD system went live and the automated system struggled to satisfy its objectives resulting in ambulances being scheduled inefficiently. The system performed similarly on the 27th October 1992 and the system was finally switched off after it crashed due to a malfunctioning fail-over mechanism on the 4th November 1992. Following this, a public enquiry was performed as it had become one of the highest profile IT failures in the UK.

The LASCAD project was revitalised by newly appointed management. Rather than pursuing the same approach as the LASCAD90 & LASCAD92 failures, they opted for a radically different approach. The new project (called LASCAD96 in this thesis) comprised a non-time pressured in-house development. A COTS solution was evaluated, but rejected, and a participative approach utilising prototyping was adopted to generate user participation and ownership (Fitzgerald & Russo, 2005). In contrast to LASCAD92, LASCAD96 invested plentiful resources in testing the system and training users. Releases of the software were delayed in situations where the user-base was not convinced of its capabilities. The new system went live on the 17th January 1996 and after a week of successful operation the staff moved into a new purpose built control room. The initial system was extremely simple and improvements were released in small increments. By September 1996 more radical enhancements were being accepted by the user-base resulting in a jump in productivity from 38%-60% of calls being despatched in 3 minutes (Fitzgerald & Russo, 2005).

6.5.1.2 Hypotheses

The threats identified using conflict analysis were used to test two hypotheses with the overall aim of illustrating the validity and reliability of approach.

[H1] The causes of the failure identified in other analyses map onto threats identified by the conflict analysis

If hypothesis one is supported then this provides an indication that conflict analysis produces reliable and valid output. This is because if the threats identified by other approaches map onto those of conflict analysis it confirms that the approach is consistent with the findings of other valid and reliable approaches to identifying threats. The hypothesis was tested by identifying the causes of the LASCAD92 failure as identified by the Official Inquiry (and supporting academic literature) and mapping these on to relevant threats identified by conflict analysis.

[H2] Threats identified in the failed LASCAD92 project were appropriately mitigated/partially mitigated in the successful LASCAD96 project

If hypothesis two is supported by the results then this corroborates that the kinds of threat identified by conflict analysis have a significant impact on the behaviour of systems composed of multiple agents with differing interests. Hypothesis two was tested by identifying if the changes to practices identified by (Fitzgerald & Russo, 2005) in their case-study of the successful turn-around (LASCAD96) mitigated or partially mitigated each threat identified by conflict analysis.

6.5.1.3 Method

To analyse the LASCAD92 and LASCAD96 systems the process described in section 6.4 was adopted. Data collection consisted identifying and reviewing publically available documents on the LASCAD92 and LASCAD96 projects. Our sources of data were the official report of the inquiry into the London Ambulance Service (Page et al., 1993), and from academic papers (P. Beynon-Davies, 1995; Paul Beynon-Davies, 1999; Finkelstein & Dowell, 1996; Fitzgerald & Russo, 2005; Hougham, 1996). Each of the data sources were used to establish and corroborate the LASCAD's:

- Key stakeholders,
- Stakeholder responsibilities with respect to ambulance dispatch system
- Resources required to fulfil responsibilities
- Dependencies between responsibilities
- Stakeholders' capabilities, interests and histories of conflict

The information acquired was consolidated and represented visually using responsibility models – see Figure 6.1 and Figure 6.2 over page. Information relevant for conflict analysis was used to populate conflict analysis checklist templates – see Appendix A for an example template.

6.5.1.4 Responsibility Models

After data collection, responsibility models of each system were developed to aid in threat identification. The LASCAD92 system is represented in Figure 6.1 below. One can observe that the system has four key stakeholders. The LAS executives, whom are responsible for meeting ORCON requirements, the

LASCAD management, whom are responsible for monitoring and managing ambulance dispatch, the Control Room staff, whom are responsible for receiving calls, entering the calls into the LASCAD system and the dispatch of ambulances. And the Ambulance Crew, whom is responsible for acknowledging dispatch requests, attending incidents and transporting patients to hospital. We can also observe that the system depends upon multiple resources. The LAS executives are dependent upon the LAS management fulfilling their responsibility to monitor and manage ambulance dispatch. The LAS management depend upon the LASCAD software system generating 'incident statistics' to fulfil this obligation. The Control Room Staff depend upon a telephone system and the LASCAD software system. The Ambulance Crew depends upon the ambulance vehicle system for dispatch requests. One can also see that many of these resources are not independent but are mostly provisioned by either the LASCAD software system or the ambulance vehicle system. This signals to the analyst that resources will have shared threats e.g. A LASCAD software, or Ambulance Vehicle system failure, may result in multiple responsibility fulfilment failures.

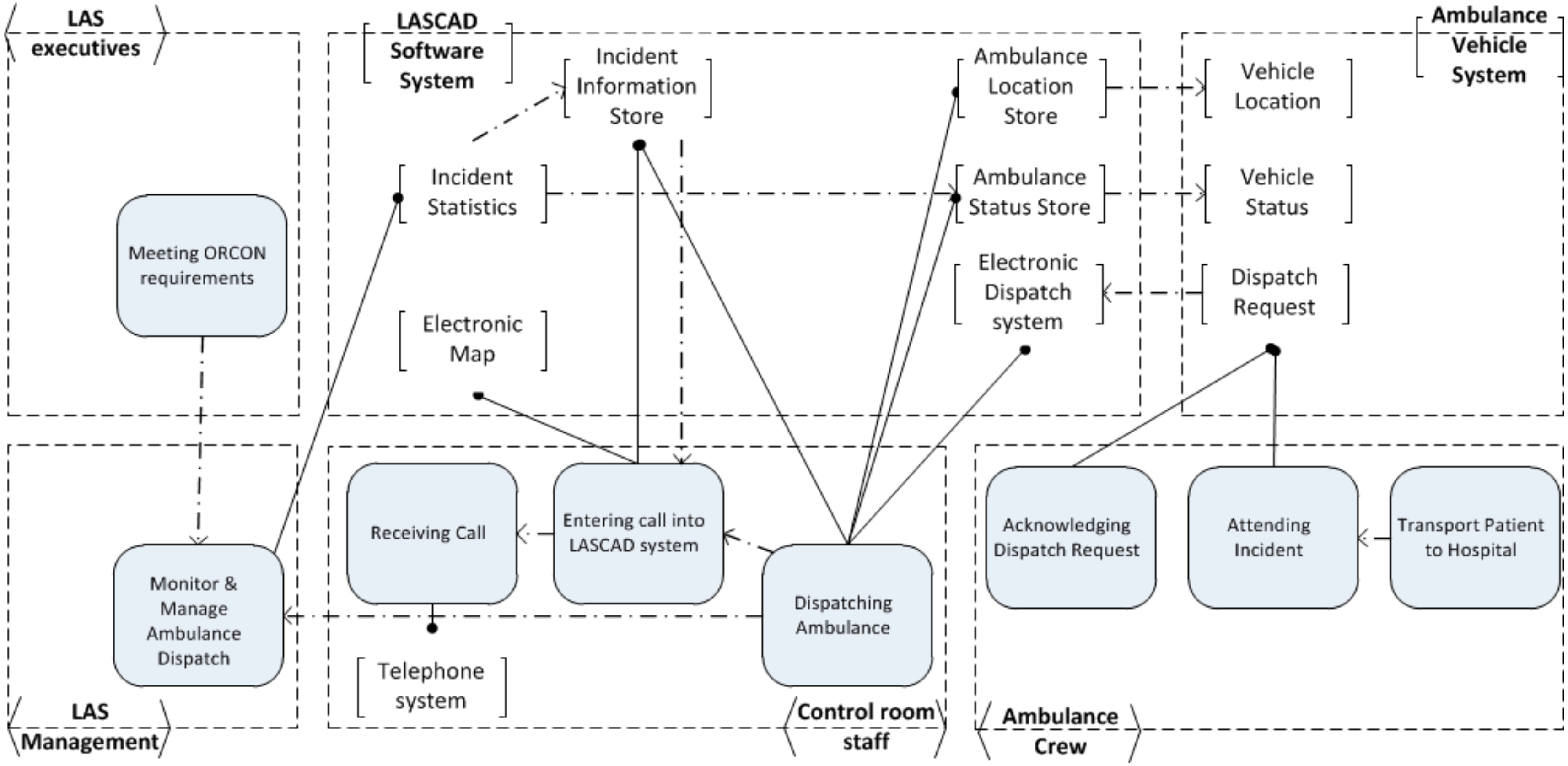


Figure 6.1 - LASCAD92 system – Go Live

The LASCAD96 system is represented in Figure 6.2 below. Similarly to the LASCAD92 model we can see that there are four key stakeholders, the LAS executives, the LAS management, the control room staff and the ambulance crew. Crucially we can see that the allocation of responsibilities is subtlety different in the LASCAD96 system in comparison to the LASCAD92 system. In LASCAD96 it is the human operators who are responsible for dispatching the ambulances and for communicating the dispatch request to the ambulance crew via radio or telephone. In the LASCAD92 system, although the control room staff are responsible for dispatching ambulances, it was the software that performed this action via a data-link to the ambulance vehicle system, which obfuscates whether or not the dispatch request was accepted since there was no human interaction. Also crucially in the LASCAD96 system, control room staff are responsible for updating the status of incidents via radioing ambulance crew and asking for situation updates if they are unsure of whether progress is being made. This means that all the status and location data in the LASCAD96 software system is gathered by a human and tacitly sense checked prior to being entered into the software system. In contrast, the LASCAD92 software was designed to automatically gather vehicle location and status from each ambulance vehicle system. This minimisation of human interaction between control room and ambulance crews threatens to cause confusion should events occur that the system has not been designed to take into account, e.g. if ambulance crews attend an incident in vehicles other than those specially assigned by the software system, or if ambulance crews swap incidents because of differing levels of expertise or road conditions.

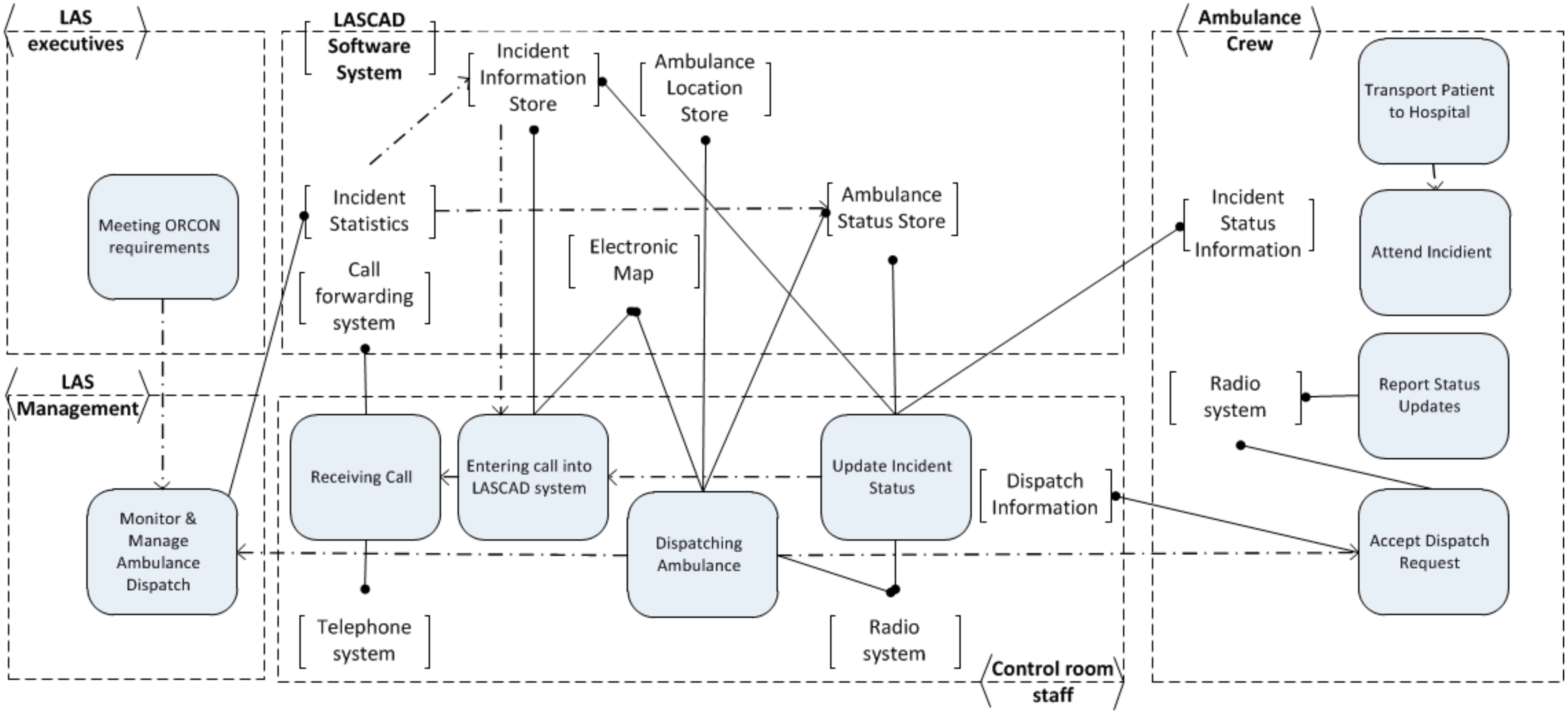


Figure 6.2 - LASCAD96 system – Go Live

The above graphical representations of LASCAD92 and LASCAD96 were used to guide conflict analysis. For each agent in the diagram a conflict analysis checklist template was completed. The responsibility models assisted in this process by prompting the analyst to consider each responsibility, its required resources, and to consider inter-agent conflicts (conflict that occur when one agent depends on another).

Since this chapter focuses on conflict analysis the threats identified are purely conflict related threats. Threats related to responsibilities being unfulfilled for reasons other than conflict will not be analysed here. Of course such an analysis could have been performed. For instance, one may trivially identify that if the ambulance crews' radios, or their ambulance vehicle system, becomes inoperable due to radio interference then they will be unable to accept dispatch requests, which will have implications for control room staff fulfilling their responsibility to dispatch ambulances, which will ultimately have consequences for the LAS executives meeting ORCON requirements. However since this capability was demonstrated in the previous chapter, we will focus on the harder challenge of identifying threats resulting from intra-agent and inter-agent conflict.

6.5.1.5 Identified Threats

In total 24 potential conflicts were identified. These included new resources being perceived to interfere / impede agents' fulfilment of responsibilities, new resources requiring capabilities that the agents' do not possess, new resources interfering with agents' values and satisfaction, new resources increasing the individual agents' workloads and thus interfering with their satisfaction and ability to fulfil responsibilities. Potential inter-agent conflicts were also identified, for instance, the distribution of system benefits vs. drawbacks being distributively unjust, the procedure by which the change was implemented being procedurally unjust. A full list of threats may be explored in Appendix B. In this section for the purposes of illustration we will discuss a subset of these threats.

Eight conflict related threats were identified and associated with the LAS management. Perhaps the most salient threat clauses were:

- LAS management perceive that their jobs are under threat if they appear uncooperative (Condition). The LAS management will be reluctant to report negative information about the system to LAS executives for fear of reprisals (Threat). LAS management will therefore suppress negative information about the project for fear of losing their job (Consequences).
- LAS management perceive that LAS control room staff and ambulance staff intend to obstruct the new system because of poor worker-management relations (Condition). The LAS management may ignore negative feedback about the system as politically motivated behaviour

(threat). LAS management may ignore valid criticism of the system resulting in it being unfit for purpose when deployed (consequence).

- LAS management view the imposition of the LASCAD software system as procedurally unjust because LAS is an exceptional case due to the size of its region (Condition). LAS managers uncooperative or with hold information during the development (Threat). LAS software system may be unfit for purpose when deployed (Consequence).

Eight conflict related threats were identified and associated with the control room assistants. These included:

- Control room staff viewed the imposition of the LASCAD software system as distributed unfairly because their jobs will radically change but in return they do not get better working conditions or increases in pay (Condition). Control room staff will obstruct the development or attempt to sabotage the system (Threat). LAS software system may be unfit for purpose when deployed (Consequence).
- Control room staff perceives the LASCAD software as removing satisfying work (Condition). Control room staff will obstruct the development or attempt to sabotage the system (Threat). LAS software system may be unfit for purpose when deployed (Consequence).
- Control room staff perceives the LASCAD software as degrading their ability to meet their values of rapid patient care (Condition). Control room staff may obstruct the development or provide negative feedback about the (Threat). LAS software system may be unfit for purpose when deployed or may require significant rework prior to being fit for purpose (Consequence).

Eight conflict related threats were identified and associated with ambulance crew. These included:

- Ambulance crew are inadequately trained (Condition). Ambulance crew will not have the capability of fulfilling their responsibilities to acknowledge dispatch requests and report incident attendance leading to dissatisfaction (Threat). Control room staff may not be able to fulfil their responsibility to dispatch ambulances as their requests go unacknowledged (Consequence).
- LASCAD system requires ambulance crew to attend incidents as decided by the software ambulance selection algorithms (Condition). Following the system may interfere with ambulance crews' values of rapid patient care and may disobey the system if instructions go against their judgement (Threat). Ambulances may not accept dispatch requests, or may attend a dispatch request in a different vehicle to the one scheduled by the system (Consequence).
- LASCAD system requires ambulance crew to attend incidents as decided by the software ambulance selection algorithms (Condition). Following the system may reduce ambulance crews' satisfaction since it removes the autonomy of crews to use their local knowledge and

experience when accepting/attending incidents (Threat). Ambulances may not accept dispatch requests, or may attend a dispatch request in a different vehicle to the one scheduled by the system (Consequence).

These example threats are intended to provide a representative sample of threats to the LASCAD system because of the system conflicting with agent's responsibility fulfilment, satisfaction or notions of fairness.

6.5.1.6 Hypothesis Testing

In this section we test two hypotheses with the aim of establishing whether conflict analysis may be a useful approach for identifying threats that affect the behaviour of systems.

Hypothesis 1: threats identified by conflict analysis map onto the causes of failure identified in other analyses

The first hypothesis will be tested by verifying whether the officially identified causes of the failure of LASCAD92, and those identified by academic papers map onto the threats identified using conflict analysis. If the causes do map onto threats identified by conflict analysis then it can be claimed that conflict analysis identifies threats that can have adverse effects on system behaviours.

When I performed the mapping exercise all but one cause could be mapped to threats identified using conflict analysis. A full table of mappings may be found in Appendix C. Hypothesis 1 is therefore supported by these results, and this corroborates that conflict analysis captures threats captured by existing approaches and therefore provides an indication of the reliability and validity of the method.

An example of a mapping between an officially identified cause and a conflict analysis identified threat is as follows. The report of the inquiry into the LAS identified that external pressure to achieve results resulted in insufficient time being allowed for developing and testing of the extremely complex technical solution. This maps to the inappropriate mitigation of threat [M2] using coercion rather than the provisioning of time and resources. Threat [M2] is that LAS management may resist the IT system if they perceive that they will be given inadequate time and resources for it to function. This threat was mitigated by coercion (applying pressure to managers) they preventing them for resisting/conflicting with the system until they perceived it to be fit for purpose.

Another example of a mapping is between the identified cause that there was a lack of disciplined technical approach. This can be mapped onto the mitigation of threat [M3] by coercion (applying pressure) rather than ensuring management have appropriate capabilities and knowledge. Threat [M3] is the LAS management will resist/conflict with the system if they cannot, or are unwilling, to develop skills to manage the system or its development. Again rather than providing an environment where LAS management could speak out about problems, coercion was used to discourage any behaviour that could be perceived as obstructive to the system being deployed as quickly as possible.

Hypothesis 2: the majority of conflict threats identified in the failed LASCAD92 project are unmitigated, whilst the majority of conflict threats are appropriately mitigated/partially mitigated in the successful LASCAD96 project.

The second hypothesis will be tested by verifying whether the majority of threats identified by conflict analysis were unmitigated in LASCAD92 but appropriately mitigated in LASCAD96. If this is the case then it may be claimed that the kinds of threat identified by conflict analysis can have adverse effects on system behaviour unless appropriately mitigated.

The conflict analysis found 24 conflict threats associated with the LASCAD 92 and 96 projects. Of these 24 threats, 2 were appropriately mitigated or partially mitigated during LASCAD 92. However during LASCAD 96, 23 of the 24 threats were appropriately mitigated or partially mitigated. Since LASCAD 96 was considered a success this suggests that the conflict threat factors if left unmitigated influences the behaviour of systems composed of multiple agents with differing interests. Therefore hypothesis 2 is supported, as most of the threats identified in the LASCAD92 project were mitigated/partially mitigated in the successful LASCAD96 project

The two appropriately mitigated stakeholder threats during the LASCAD 92 project were: [M1] the threat that LAS Management may resist the IT system as they perceive the changes in working practices to degrade their chances of meeting ORCON (a standard for monitoring ambulance service performance) targets; [M4] the threat that LAS management may resist the IT system as they perceive dependency on new, perhaps unreliable, technology to degrade their chances of meeting ORCON targets. Both of these threats were mitigated by the fact that management was convinced that radical technology was the solution to their problem.

The unmitigated stakeholder threat during the LASCAD 96 project was [M6] the threat that LAS management may view Control Room assistants or ambulance crew as being obstructive when providing negative feedback about the system due to poor past relations. There is no evidence to suggest that this threat materialised as all evidence reviewed suggests that management were open to all bottom-up feedback to the extent that they delayed a phase of the deployment until staff members were confident in its use.

6.5.1.7 Lessons Learned

The LASCAD 92/96 project emphasises the importance of agent conflict and its adverse effect on the behaviour of systems composed of multiple agents with differing interests.

[1] Resistance comprises stakeholders providing feedback on how they perceive a system to impact their local environment and therefore addressing their perceived threats is valuable as it facilitates good fit between their local environment and the system and the changes it brings about.

[2] The majority of stakeholder threats can be appropriately mitigated, or partially mitigated, by senior management demonstrating that they are willing to invest the resources it takes to get a project done well and that they are open and respond to continuous consultation/feedback from all stakeholders.

[3] Sources of continuous consultation/feedback include: Up-front consultation; ongoing drop-in sessions; User acceptance testing where users can delay go-live if unhappy.

[4] Software and system designers should be mindful of stakeholders' values, satisfaction, and status. One particularly important area is to avoid fully automating user decision-making and instead supporting the user to make better decisions. This is beneficial as it reduces the threat of removing satisfying or status-granting aspects work whilst simultaneously enabling the user to incorporate their local knowledge or expertise.

[5] Stakeholder threats that are mitigated via coercion tend dampen feedback loops between stakeholders resulting in poor communication and ultimately a project that is not a good fit with its environment.

6.5.2 Cloud Adoption in an Oil/Gas Sector SME

The aim of this case study is to demonstrate the use of conflict analysis to identify threats to a system-of-systems. The case study organisation is a UK based SME that provides bespoke IT solutions for the oil & gas industry. It consists of around 30 employees with offices in the UK and the Middle East. It has an organisational structure based on functional divisions (e.g. administration, engineering and support).

The migration use-case comprised determining the feasibility of migrating one of the organisation's primary service offerings (a quality monitoring and data acquisition system) to Amazon EC2 – an infrastructure-as-a-service offering from Amazon Web Services. The following is an anonymised description of the situation: *Company C* is a small oil and gas company who owns some offshore assets in the North Sea oilfields. Company C needed a data acquisition system to allow them to manage their offshore operations by monitoring data from their assets on a minute-by-minute basis. Company C's assets rely on the production facilities of *Company A* (a major oil company), therefore the data comes onshore via Company A's communication links. Company C does not have the capabilities to develop their own IT systems, hence they outsourced the development and management of the system to *Company B*, which is an IT solutions company with a small data centre. Figure 6.3 provides an overview of the system, which consists of two servers:

1. A database server that logs and archives the data coming in from the offshore platform into a database. A tape drive is used to take daily backups of the database, the tapes are stored off-site.
2. An application server that hosts a number of data reporting and monitoring applications. The end users at Company C access these applications using a remote desktop client over the Internet.

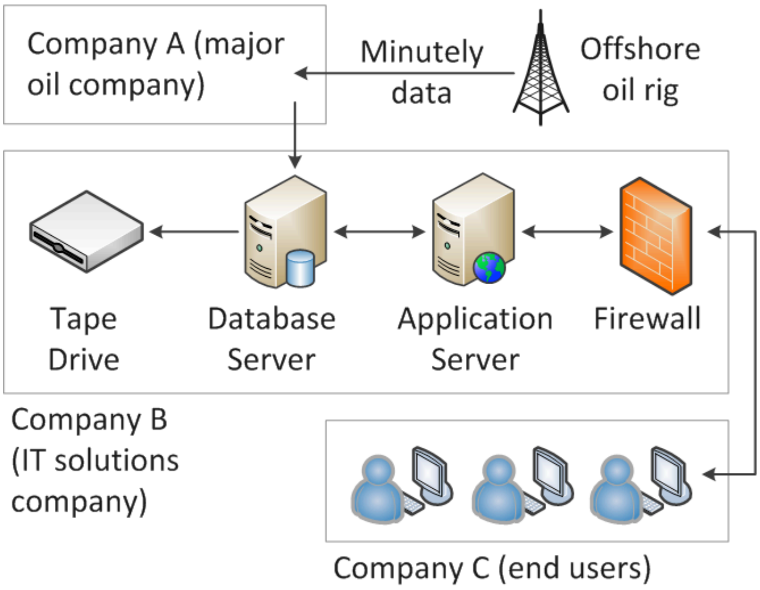


Figure 6.3 - Oil & Gas Case study: System Overview

The system infrastructure was deployed in Company B's data centre and went live in 2005. Since then, Company B's support department has been maintaining the system and solving any problems that have risen. This case study investigated how the same system could be deployed using the EC2 cloud offerings of Amazon Web Services. Figure 6.4 provides an overview of this scenario, where Company B deploys and maintains the same system in the cloud.

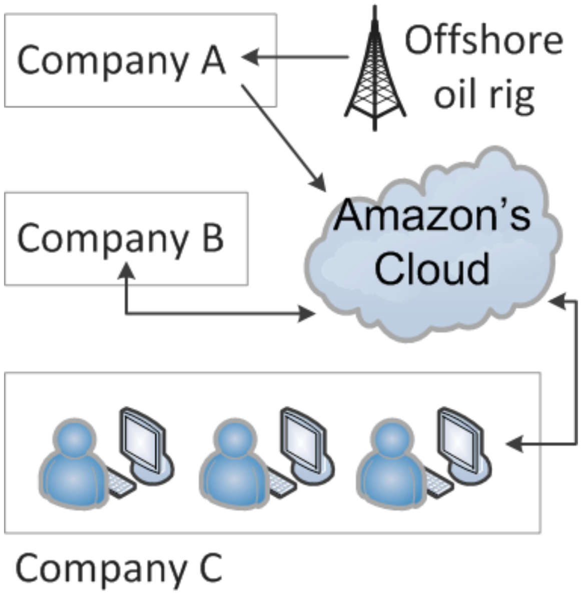


Figure 6.4 - System deployed in the cloud

A responsibility modelling approach was used to identify threats related to the proposed change. Data was collected via the use of five semi-structured interviews with staff members from different functions within the organisation (e.g. Technical manager, project manager, business development manager, systems engineers, support technicians) and supported with documents where available. Data was analysed using the conflict analysis approach by one researcher and corroborated by a second to help ensure reliability of finding.

Responsibility analysis was performed in a manner similar to the LASCAD case study. Interview data was used to establish and corroborate the proposed cloud systems:

- Key stakeholders,
- Stakeholder responsibilities with respect to the data acquisition and monitoring system
- Resources required to fulfil responsibilities
- Dependencies between responsibilities
- Stakeholders' capabilities, interests and histories of conflict

The information acquired was consolidated and represented visually using responsibility models – see Figure 6.5 and Figure 6.6 overleaf. Information relevant for conflict analysis was used to populate conflict analysis checklist templates.

The 'as-is' responsibility model (Figure 6.5) describes the existing in-house data acquisition and monitoring system. One can see that the system is composed of seven agents. Company A, whom are responsible for providing the offshore gateway that transports data onshore to company B's data centre. Company C, whom are responsible for paying for the metering system and paying for the metering support contract with Company B. We can see that Company B is represented as five human agents, whom comprise the:

- Support manager who is responsible for the provision of the metering service and the metering service technical support;
- Customer relationship staff, which are responsible for responding to customer queries with respect to technical matters and also service provision agreements and billing;
- Metering system engineers, who are responsible for support and maintenance of the hardware, operating system, database, application, fallback and recovery, back-ups and network support and maintenance;
- Sales and marketing staff, who are responsible for creating awareness of products in the marketplace, identifying potential customers and matching their needs to products/services, convincing customers that products/services meet their needs;
- Finance / Business development staff, who are responsible for managing expenditure, paying for the offshore gateway, managing cash-flow, managing income, and managing market share.

We can also observe that the system depends upon multiple resources. The in-house metering system, documentation for troubleshooting and configuration of the metering system, redundant hardware for fallback, back up media, the customer relationship and billing system, the offshore gateway, knowledge of in-house service offerings and means of payment. We can also observe dependencies between responsibilities that span organisational boundaries. For instance for the support manager to fulfil his responsibility of provisioning and maintaining the metering system he is dependent upon Company A fulfilling their responsibility of provisioning and maintaining the offshore gateway.

The ‘to-be’ responsibility model (Figure 6.6) describe the proposed the EC2 cloud-based data acquisition and monitoring system. This system is composed of eight agents. Company A is responsible for provisioning and maintaining the offshore gateway, AWS is responsible for provisioning EC2 virtual machines and AWS gold support, Company C is responsible for paying for the metering system and the metering system support contract. Company B as represented by its five human agents are responsible for responsibilities similar to those in the ‘as-is’ system.

The most notable differences between the ‘as-is’ system and the ‘to-be’ system are extent that company B’s responsibility fulfilment is dependent upon resources that are outside their organisational boundary and a reduction in the number and nature of the metering system engineers responsibilities. One may now observe that the Support manager’s responsibility to provision the metering system is dependent AWS EC2 virtual machines, AWS support staff and metering system engineers having access to EC2 documentation (which AWS provisions but does not take responsibility for). We will now use conflict analysis to get a better understanding of these changes by attempting to understand their implications for each agent’s interests.

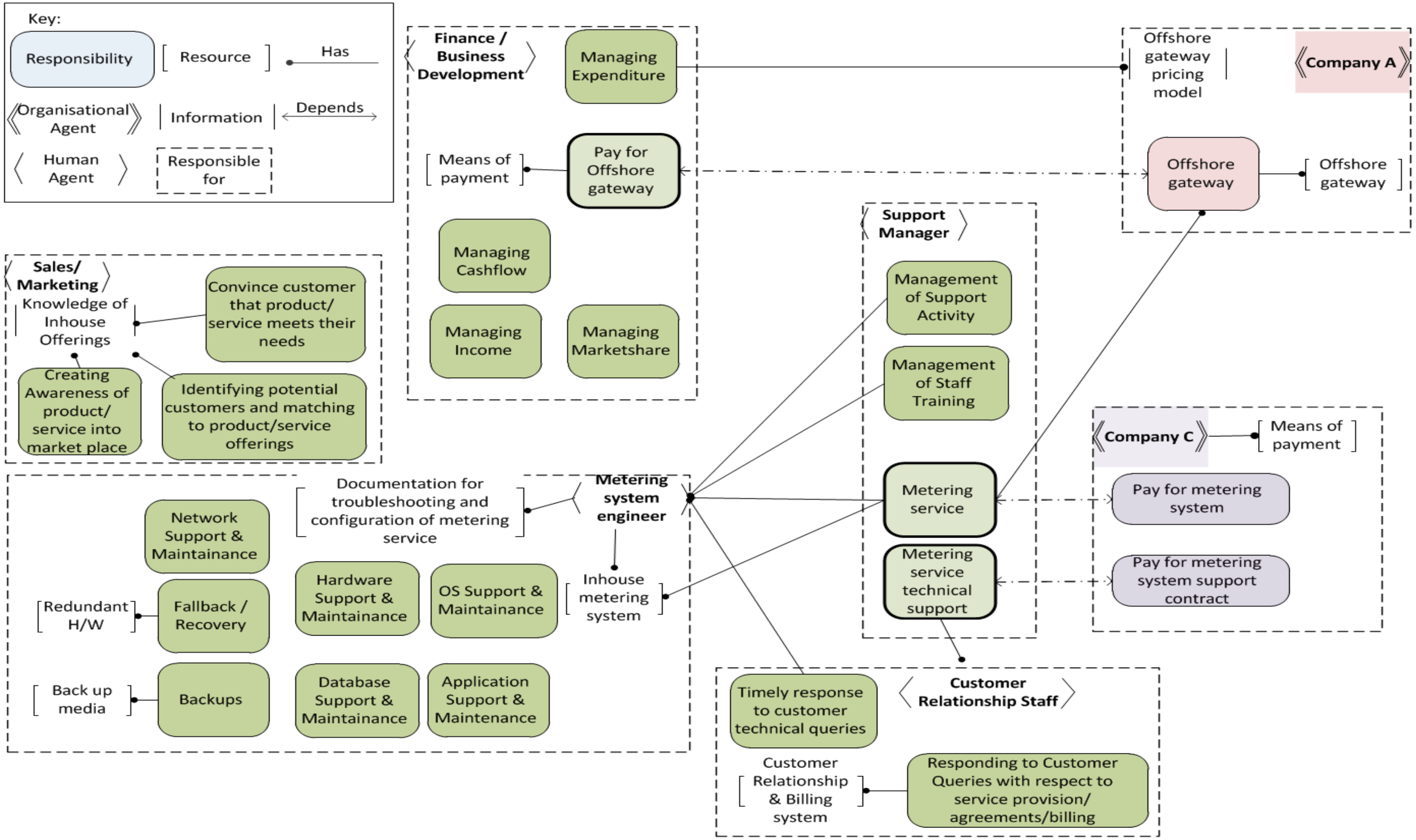

Key:
Responsibility
Resource
Has
Organisational Agent
Information
Depends
Human Agent
Responsible for
Finance / Business Development
Managing Expenditure
Means of payment
Pay for Offshore gateway
Managing Cashflow
Managing Income
Managing Marketshare
Offshore gateway pricing model
Company A
Offshore gateway
Offshore gateway
Sales/ Marketing
Knowledge of Inhouse Offerings
Convince customer that product/ service meets their needs
Creating Awareness of product/ service into market place
Identifying potential customers and matching to product/service offerings
Support Manager
Management of Support Activity
Management of Staff Training
Metering service
Metering service technical support
Company C
Means of payment
Pay for metering system
Pay for metering system support contract
Documentation for troubleshooting and configuration of metering service
Metering system engineer
Inhouse metering system
Network Support & Maintainance
Redundant H/W
Fallback / Recovery
Hardware Support & Maintainance
OS Support & Maintainance
Back up media
Backups
Database Support & Maintainance
Application Support & Maintenance
Timely response to customer technical queries
Customer Relationship Staff
Customer Relationship & Billing system
Responding to Customer Queries with respect to service provision/ agreements/billing

Figure 6.5 - As-is responsibility model of the system

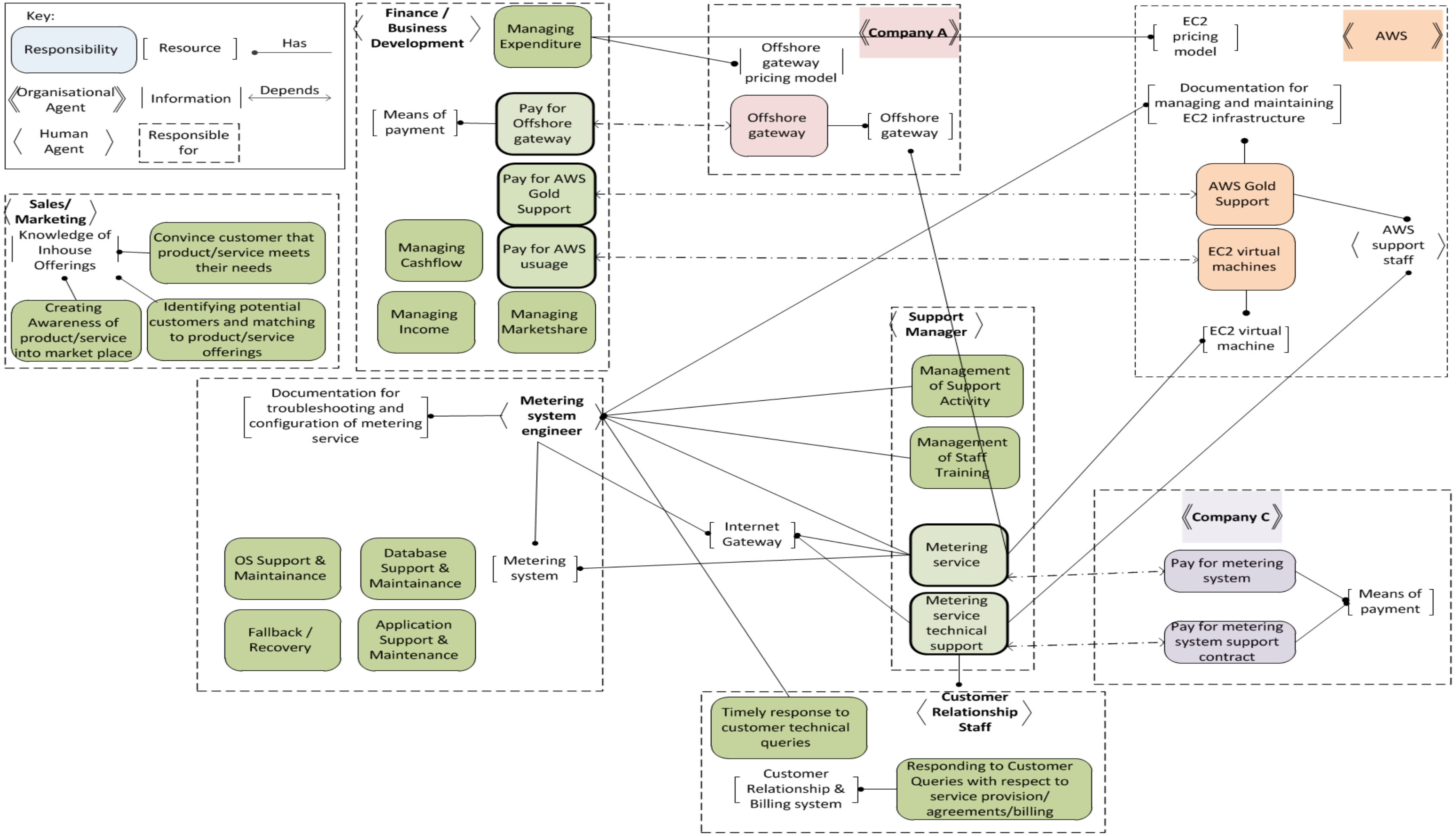

Key:
Responsibility
Resource
Has
Organisational Agent
Information
Depends
Human Agent
Responsible for
Finance / Business Development
Managing Expenditure
Means of payment
Pay for Offshore gateway
Pay for AWS Gold Support
Managing Cashflow
Pay for AWS usage
Managing Income
Managing Marketshare
Company A
Offshore gateway pricing model
Offshore gateway
Offshore gateway
EC2 pricing model
AWS
Documentation for managing and maintaining EC2 infrastructure
AWS Gold Support
EC2 virtual machines
AWS support staff
EC2 virtual machine
Sales/ Marketing
Knowledge of Inhouse Offerings
Convince customer that product/service meets their needs
Creating Awareness of product/service into market place
Identifying potential customers and matching to product/service offerings
Documentation for troubleshooting and configuration of metering service
Metering system engineer
OS Support & Maintainance
Database Support & Maintainance
Metering system
Fallback / Recovery
Application Support & Maintenance
Support Manager
Management of Support Activity
Management of Staff Training
Internet Gateway
Metering service
Metering service technical support
Company C
Pay for metering system
Means of payment
Pay for metering system support contract
Customer Relationship Staff
Timely response to customer technical queries
Customer Relationship & Billing system
Responding to Customer Queries with respect to service provision/ agreements/billing

Figure 6.6 - To-be responsibility model of the system

6.5.2.2 Conflict Analysis

Conflict analysis was performed by understanding the implications of the system changing from the ‘as-is’ configuration to the ‘to-be’ configuration. This comprised:

1. Identifying changes in what responsibilities an actor is expected to perform;
2. Identifying changes in how responsibilities will be performed;
3. Identifying changes in agents required time, resources or capabilities to ensure appropriate fulfilment of responsibilities;
4. Identifying potential risk of not having required time, resources or capabilities;
5. Identifying risk of interference between required activity, individual values, status and satisfaction;
6. Identifying risk of multiple incompatible responsibilities;
7. Identifying risk of incompatibilities between agents e.g. histories of conflict;
8. Identifying risk of perceived distributive injustice or procedural injustice.

Risk clauses were identified by considering *conditions* that could result in a *threat* to responsibility fulfilment, and the *consequences* of this for the fulfilment of other responsibilities. For example from the perspective of the support manager it was identified that if AWS support staff are not forthcoming in resolving support requests (condition), then the support manager’s responsibility of providing metering service technical support may not be fulfilled (threat), resulting in possible SLA violations and financial loss. Another salient example is that if the support manager perceives the system to downsize his department (reduce his number of required resources) because of the outsourcing of network support and back-ups (condition), the support manager may conflict with the project (threat), resulting in a system that does not fulfil metering system SLAs. Another salient example is that if support manager perceives that he lacks required resources because of lack of staff capabilities and experience in cloud systems (condition), the support manager may conflict with the project (threat), resulting in a system that does not fulfil metering system SLAs. See Appendix D for a complete list of risks.

From the perspective of the support engineers a number of conflict related risks were also identified. For instance, if the system engineers perceive that the removal of hardware, network and backup responsibilities is a threat to their working hours or job satisfaction (condition), then support engineers may conflict with the system (threat), resulting in a system that does not fulfil metering system SLAs. Another risk is that, if system engineers perceive that the dependency on EC2 hardware introduces additional satisfying work e.g. contacting Amazon support rather than fixing in house issues (condition), then support engineers may conflict with the system (threat), resulting in a system that does not fulfil metering system SLAs. See Appendix D for a complete list of risks.

Risks were also identified from the perspective of non-technical staff. For instance, if sales and marketing staff do not have the capability to understand how a cloud solution is useful to customers (conditions), then they may conflict with the system (threat), resulting in poor sales of the cloud-based offerings (consequence). From the finance and business development perspective, if finance staff perceive that the system exposes the organisation to significant fluctuations in bandwidth, processing and storage costs (condition), then variability may threaten the finance staff's responsibility to manage cash flow, thus resulting in additional operational risk. From the perspective of customer relationship staff, if customer relationship staff do not have appropriate knowledge and capability to understand cloud style services (condition), then there is a threat to the timely response of customer queries (threat), as a result SLAs may be breached resulting in financial liability or degradation of customer service resulting in dissatisfaction (consequence).

6.5.1.3 Lessons Learned

The following lessons were learnt from the case study:

- Conflict derived risks can outweigh the substantial financial opportunities of an IT project.
- The systematic structuring of thinking and data to reason about conflict sensitises analysts to unexpected, but significant, factors that contribute to determining the feasibility of a project.
- Localising the negative consequences of conflict to specific stakeholder groups is a powerful approach to getting a handle on the impact of agents' differing interests on system behaviour.

6.6 Conclusion

This chapter demonstrated that conflict analysis promises to be a reliable and valid approach to identifying threats to systems composed of cooperating agents. This was demonstrated through two case studies. The LASCAD case study supports that conflict analysis is a reliable and valid approach to identifying risks. Validity was indicated by the fact that the risks identified by independent analyses of the LASCAD92 project mapped onto risks identified using conflict analysis. Validity was also indicated by the fact that the successful LASCAD96 had far fewer unmitigated risks in comparison to the LASCAD92 project. A limitation of these findings is that the author was aware that the LASCAD92 should have fewer mitigated risks than the LASCAD96 system; therefore unintended bias may be present in the results.

The oil/gas sector SME case study indicated the three fold value of conflict analysis: firstly its ability to sensitise practitioners to agent conflict by systematically structuring a practitioner's thinking and data in-order to be able to reason about it; secondly that the approach encourages thinking about management support, user support and adequacy of resources which are three aspects of project performance that are empirically demonstrated to have a significant effect on IT project performance (Xia & Lee, 2004); thirdly that the approach has the ability to trace the negative consequences of conflict to specific stakeholder groups and present the information at a granularity that practitioners may be able to utilise to inform practice.

The conclusions of this chapter are limited by the usual limitations of case study approaches. Case studies are descriptive and conclusions about causes and effects cannot be safely drawn since they do not comprise controlled laboratory conditions. Also, case studies may not be representative of a general population and due to the nature of the research topic the data is collected from human subjects so it is potentially subject to bias. Every effort was made to control for these shortcomings by using multiple sources to corroborate information where possible and using multiple case studies from contrasting sectors and technology types.

7. Structural-Intentional Responsibility Modelling for Troubleshooting a Problematic Enterprise-scale System

7.1 Introduction

Enterprise-scale systems are systems intended to serve the needs of multiple human agents whom work within the same organisation but may hold differing priorities due to differing responsibilities, departmental loyalties and so on. The aim of this chapter is to demonstrate that the modelling of responsibilities aids the troubleshooting of enterprise-scale system behaviour. This is shown through the application of structural-intentional responsibility modelling to a case study of a multi-national systems engineering firm with a problematic enterprise document management (EDM) system.

Structural-intentional responsibility modelling augments responsibilities with intentional and structural notions from activity systems theory to provide a modelling framework with greater granularity of analysis. This provides an analyst with a troubleshooting framework to identify a variety of potential conflict, also referred to as tensions, between interacting agents. Tensions can be thought of in terms of an agent's behaviour being pulled in incompatible directions. For instance, an agent may be simultaneously encouraged to always fulfil a responsibility by following a defined process but also encouraged to reduce the time it takes to fulfil a responsibility. The agent is therefore conflicted whether to behave in a manner that follows the defined process, or in a manner that does not adhere to the process but reduces fulfilment time.

Our use of structural-intentional responsibility modelling to troubleshoot a problematic enterprise document management system resulted in the identification of 14 software and system usability issues and a set of six structural and intentional incompatibilities. This enabled the identification of four vicious circles that we hypothesise contributed to the system being perceived to behave in a problematic manner by engineering management. These insights were used to suggest how the behaviour of the agents and the EDM artefact could be changed so that the problematic behaviours of the EDM system could be ameliorated. Overall the case study supports the claim that structural-intentional responsibility modelling enables the analysis of the behavioural properties of enterprise-scale systems.

This chapter is structured such that section 2 provides specific background and motivation for the development of structural-intentional responsibility modelling. Section 3 introduces the case study design, data collection methods, the framework used to analyse the data, and the case study organisation. Section 4 describes the case study findings, first highlighting software and system level issues prior to suggesting that the behaviour of the system is most coherently accounted for through the analysis of structural and intentional issues. In section 5 we discuss the case study findings with respect to our aim of analysing the behaviour of enterprise systems and how structural-intentional responsibility modelling contrasts and complements other approaches I* and SSM. In section 6 we conclude that structural-intentional responsibility modelling is a viable candidate as a scalable engineering technique for analysing and troubleshooting enterprise-systems post-implementation.

7.2 Social Analysis to Inform System Implementation

Information system development is the process of conceiving, analysing, designing and implementing an information system (Avison & Fitzgerald, 2006). The process of implementation comprises the deployment, adoption and routinisation of a system within its environment. Implementation can be challenging because of unforeseen problems. For instance, users may not be interested in using a system because they perceive it not to help them fulfil their responsibilities, or that it creates them additional work (Doherty & King, 2005). In order to minimise these types of issue, methods of social analysis such as Coherence (Viller & Sommerville, 1999, 2000), ETHICS (Mumford, 1995), Multiview (Avison et al., 1998) and I* (E. S. K. Yu, 1997) have been developed to inform the analysis and design of systems.

There is recognition within the socio-technical systems community that practitioners' needs are not being met by the above analysis and design methods (Bygstad, Nielsen, & Munkvold, 2010). This is motivated by their lack of use by industry and recognition of the 'design fallacy'. The design fallacy is the assumption that the primary solution to meeting users' needs is to develop ever more sophisticated social analyses to inform the design of systems (Stewart & Williams, 2005). Qualitative studies of system development show that implementation outcome is not only influenced by a system's design but is also strongly influenced by groups of people ('technology mediators') shaping the familiarisation of a system and the ability of organisational and technical infrastructure to facilitate the adaptation of the system and associated work practices to users circumstances during implementation (Anderson, Hardstone, Procter, & Williams, 2005; Doherty & King, 2005; Orlikowski, Yates, Okamura, & Fujimoto, 1995; R. Williams, Stewart, & Slack, 2005).

The idea that 'better design is not enough' is largely unexplored from an engineering perspective. As a result the engineering community is devoid of engineering approaches to identify socio-technical issues that may be inhibiting adaptation and familiarisation of a system. Similarly we have yet to develop knowledge of the most effective and efficient combinations of organisational and technical infrastructure (e.g. IS steering groups, user groups, wikis and so on) to facilitate adaptation and familiarisation.

Quantitative and qualitative studies of system adoption demonstrate a strong relationship between system adoption and structural-intentional elements. For instance the two best-established quantitative models of system adoption or success suggest the importance of intentional factors e.g. beliefs and intentions with respect to a system. The Delone-Mclean model (Delone & McLean, 2003) highlights the importance of the relationship between users' intentions and their beliefs about system quality, service quality and information quality. Similarly the UTAUT model (Venkatesh et al., 2003) highlights the importance of the relationship between system adoption and users' beliefs about 'performance expectancy', 'effort expectancy', 'social influence' as well as 'facilitating conditions' such as appropriate technical and organisational infrastructure to support the use of a system. Qualitative studies also indicate the importance of structural-intentional elements. For example (Anderson et al., 2005; Doherty & King, 2005; Orlikowski et al., 1995; R. Williams et al., 2005) illustrate the role of institutional rules shaping the adoption and use of technology as well as the important role of people acting as 'technology mediators'.

Since the above studies indicate a strong relationship between system adoption and structural-intentional factors, this thesis chapter proposes to model and analyse some of these factors. There has been little work specifically directed at developing engineering approaches that inform deployment and adoption, however there is an existing body of work that addresses the *troubleshooting* of problematic socio-technical systems. This work has largely come from related fields outside of socio-technical systems engineering such as ergonomics, CSCW (Computer Supported Collaborative Work) and Soft OR (Operations Research). Some of the most notable approaches include situated action (SA) (Suchman, 1987), distributed cognition (DC) (Hutchins, 1995), activity systems theory (AST) (Engestrom, 2000) and soft systems methodology (SSM) (Checkland, 1999). Of these approaches SA and DC do not analyse intentional level issues (Halloran, 2000). We decided to adopt a variant of AST because AST enables a more granular analysis of structural-intentional issues than SSM, which is limited in scope to norms, values and roles, notions that can be already captured using the notion of responsibilities in terms of obligations, liabilities and norms.

7.3 Research Design

A case study approach was selected because our aim was to test the hypothesis that 'structural-intentional responsibility analysis enables the identification of stakeholder conflicts in enterprise-scale deployments by identifying incompatibilities between structural and intentional elements'. A case study approach was deemed appropriate since our aim was to test our hypothesis in a real-world corporation with a problematic system.

The fieldwork was performed at three different sites of a multinational system-engineering corporation that we will call 'Company A'. Their main work activity comprises the design, manufacture and maintenance of specialist electro-optical components and systems. The organisation is divided into a number of functional groups that come together under a project structure to produce customer deliverables e.g. components, systems and documents. The design of components and systems is a collaborative activity involving programme managers, engineering managers and engineers and the sharing of documents is a vital aspect of this activity.

‘Company A’ deployed an electronic document management (EDM) system in the early 2000s as it was perceived by the IT director that an EDM system would be more advantageous than using shared folders on a file server to exchange documents. There was a perception that the introduction of the system would bring about greater visibility and awareness of work rather than having different teams and functions working in information silos. Within projects it was envisioned that EDM would be an up-to-date repository of all project documentation. Teams would store their documents in personal working areas and upload them to standardised locations in standardised EDM project file structures.

When we visited the organisation in 2010 the EDM was perceived by engineering management to be problematic due to “socio-technical factors”. The use of the system was mandatory so all projects had an EDM project area but the extent that documents were being uploaded from working areas to the EDM project areas varied between teams. In addition to this the use of the EDM file structure varied between teams, as did the location of files within the file structure. As our investigation unfolded it became clear that engineering management perceived the system to be problematic because teams did not use it in a “common way”.

We collected our data using 16 one-hour semi-structured interviews of the document management system’s stakeholders. Interview participants were selected on the basis of availability by a facilitator within the organisation. The interviews comprised a set of open-ended questions and a set of closed questions comprising 7-point semantic differential scales and 7-point Likert scales. A copy of our interview questions can be found in Appendix E.

Interviews were digitally recorded and transcribed when permitted. The open-ended interview questions were designed to elicit the relationship between the participant’s view of the their work (role, responsibilities, their day-to-day activities, most serious work challenges) and the deployed system (their history with the system, which responsibilities/activities the system helps them accomplish, how it does so, what problems it introduces to their work, how the system impedes their responsibilities and activities). The closed interview questions elicited the relationship between the participant and the system by exploring aspects of IT systems that are associated with intention to use (performance expectancy, effort expectancy, information quality, system quality, support quality, system usage policy) and aspects of organisational change that can lead to conflict (interfering with roles, goals, values, resources, capabilities/skills, job satisfaction, status, procedural justice, distributive justice, importance, ownership).

Dialogue mapping was then used to organise interview data into more abstract units of information. Dialogue maps were compared to the participant’s responses to closed questions to corroborate findings. Dialogue maps were compared across participants to identify themes. The ‘activity space’ framework in combination was structural-intentional responsibility models were then used to structure the findings and provide a lens for identifying tensions between different elements within the situation.

The ‘activity space’ framework (Halloran, 2000, 2001) is a framework for structuring data and identifying problematic intentional and structural aspects of a system. The framework comprises three intentional constructs comprising: mediators (tools, beliefs, skills); subjects (responsibilities) and objectives. And three structural constructs comprising: rules (formal / informal norms); community (actors involved in a situation); and the division of labour (how work is divided).

According to ‘activity space’ theory the outcome of a situation (e.g. a deployment) is brought about by interactions between actors’ behaviour(s). Each actor’s behaviour is mediated by intentional and structural elements. So, problematic situations can arise when tensions exist within and between actors’ intentional and structural elements. By understanding these tensions a situation can be modified to change the outcome. We chose this framework as it gives primacy to the interrelationships between intentional and structural aspects of a situation. Structural-intentional responsibility models were found to be useful for the purpose of visually representing and structuring the information gathered, as well as providing a means of tracing the relationships between multiple agents with incompatible responsibilities, objectives and behaviours.

7.4 Our Findings

We report our findings in three parts: the first part identifies *software* usability issues, such as user interface issues, that frustrate the use of EDM regardless of the software’s specific configuration; the second part identifies *system* acceptability issues to do with the deployed configuration fitting the existing work environment; the third part reports on the *structural-intentional* issues that frustrate the use of EDM. In the third part, EDM is viewed as a resource that mediates (enables / constrains / transforms) work activity. In contrast to the first and second parts, the issues raised will highlight underlying tensions that result in issues or challenges that impede the use of EDM in a “common way” as desired by engineering management.

7.4.1 Software Usability Issues

We found that the following aspects confounded the usability of the tool in both experienced and novice/infrequent users. The consequences of these issues were typically frustration and/or perceptions of wasted time.

Table 7.1. Aspects Detrimental to Software Usability According to Experienced Users

#	Aspects detrimental to usability according to experienced users
1	Requires a login separate from workstation login.
2	Web-based interface is slow to respond to user interaction.
	-Screen updates and file uploads are perceived to be slow or freeze.
3	Files can only be uploaded individually using the web-based interface.
4	Files are rendered poorly when viewed in using the web-based interface.
5	Search feature does not return expected results.
6	Web-based interface has screen-rendering issues when used with browsers other than Internet Explorer 6.

Table 7.2. Additional Aspects Detrimental to Software Usability According to New / Infrequent Users

#	Additional aspects detrimental to usability according to new / infrequent users
1	Menus are cluttered and there is no obvious feature prioritisation to guide novice/infrequent users.
2	Search query presentation is difficult to understand
	-E.g. Use of MIME types.
3	The 'look & feel' of the web-based interface is dissimilar to the 'drag & drop' interfaces that end-users are generally accustomed to.

End-users found that the above issues to be slightly problematic but in general they perceive them to have a minor effect on their overall productivity, job satisfaction, speed of accomplishing work activity, and effort to use EDM.

- Most participants surveyed agreed or strongly agreed that EDM takes little effort to use on their part. They reported that EDM did not *significantly* improve nor worsen their individual productivity, responses were mainly distributed around no effect, or slight positive or negative effects.
- Most participants agreed that EDM did not *significantly* slow down or speed up their speed of accomplishing activities. Responses were distributed equally between no difference, slower and faster.

Most participants reported that EDM neither favourably nor adversely affects their job satisfaction. However participants did report that the user interface does not meet their needs and is slightly problematic. And that the search facility does not meet their needs and is slightly problematic.

These mixed responses indicate that although the system has a number of frustrating or timing wasting usability issues, the majority of users we interviewed found that it did not significantly interfere with either their overall productivity or job satisfaction. These findings are perhaps surprising as management perceive the system to be problematic. This difference is explained by the fact that the extent that each team uses EDM is in accordance with their own approach and therefore they use the system in a manner that is acceptable to them (as a team) but not necessarily in a manner that is desired by management. We can conclude therefore that issues and problems are not due to software usability.

7.4.2 System Acceptability Issues

We found that the following aspects confounded the acceptability of the tool in both experienced and novice/infrequent users**.**

Table 7.3. Aspects detrimental to system acceptability

#	Aspects detrimental to system acceptability
1	EDM is perceived to be more time consuming to use for storing documents in comparison to shared drives, or personal areas, due to the software usability issues identified.
2	EDM has been configured to offer standardised folder structures however users struggle to understand where to put their documents within these structures. They perceive that there are a variety of possible locations, which makes remembering and sharing the location of a document problematic. This interpretative flexibility enables the use of EDM in contrasting and inconsistent ways.
3	EDM project areas have no built-in document registers making it difficult to establish what documents are within a project area and which are missing. Lack of a document register is seen be problematic because of inconsistent use of the standardised folder structures which makes finding files on the basis on their expected folder location impractical.
4	EDM has practical limitations on the number of files in a single folder as this can cause freezing or degrade the performance of the search facilities. This has been mitigated on the most part by using a folder structure.
5	EDM runs on servers within a 'restricted' network and so cannot be accessed by all parts of the organisation. In a number of situations this results in end-users having to use other IT resources for document sharing undermining the purpose of the EDM.

Despite the issues identified above, the participants that we interviewed reported that, in general, the use of EDM does facilitate their working practices, they are supportive of continuing investment and development of EDM, and that EDM is considered to be slightly important, or important, to their interests and responsibilities. This indicates that despite the systems shortcomings it was recognised by those that we interviewed as a valuable tool that supports work. Again these findings may be surprisingly considering that engineering management perceive the system to be problematic. However this again highlights that end-users use the system in a manner that is acceptable to them but not necessarily in a "common way" as desired by engineering management. We therefore can conclude that system usability is not the source of problems.

7.4.3 Structural-Intentional Issues

In this subsection EDM is viewed as a resource available to ‘Company A’ employees that mediates (enables / constrains / transforms) work activity. We highlight underlying organisational tensions that result in the challenges that impede the adoption of EDM. The analysis is presented by summarising our overall findings and then describing how these findings were generated using structural-intentional responsibility modelling.

7.4.3.1 Findings

Using structural-intentional responsibility modelling we identified 6 issues that interact to produce the problematic behaviour identified by engineering management.

1. Whilst analysing the objectives of the engineering managers, programme managers and engineers we identified a potential process conflict between the objectives of engineering management and those of programme managers and engineers. Essentially engineering managers objectives prioritise strategic needs to improve process in the long term but this can be incompatible with engineers’ and programme managers’ short-term objectives of meeting contractual obligations and meeting customer expectations.
2. Whilst analysing the mediators available to the engineering managers, programme managers and engineers, we identified that the different groups placed different emphasis on the value, or salience, of the EDM’s features and those of shared folders. Essentially engineering managers valued EDM’s audit and file access control features more than programme managers and engineers. This meant that perceived alternatives to EDM such as shared folders appear far less attractive to engineering managers than to engineers and programme managers, thus explaining why use of shared folders may be prevalent.
3. Whilst analysing the structure of the socio-technical system we identified that as a result of the division of labour within ‘Company A’ the EDM has not become domesticated. A split between engineers being informally responsible for supporting technical tools and corporate IT being responsible for corporate tools resulted in no party taking ownership of domestication. This has resulted in a tool whose features/capabilities are not widely understood and incorporated into daily practice.
4. Whilst analysing the usability of the EDM software we identified a number of issues frustrating the use of the software but not significantly impeding people’s productivity or satisfaction.
5. Whilst analysing the structure of the socio-technical system we identified that informal rules of ‘Company A’ suggest the existence of a ‘strong practice culture’, rather than a process culture. This means that work is

performed on the basis of norms and experience rather than following explicit rules are specified by processes.

6. Also whilst analysing the informal rules of 'Company A' we recognised the existence of an exemption culture. Employees would find justification to exempt themselves from following processes on the basis of being a special case.

The issue map below (Figure 7.1) was created to represent the relationships between the issues and aid in communicating how they interact. The issue map indicates that four vicious circles within the situation were identified to be contributing to sustaining the problematic situation. Vicious circles are particularly important to identify, as any intervention to ameliorate the situation must disrupt the vicious circles in order to change the outcome of the situation.

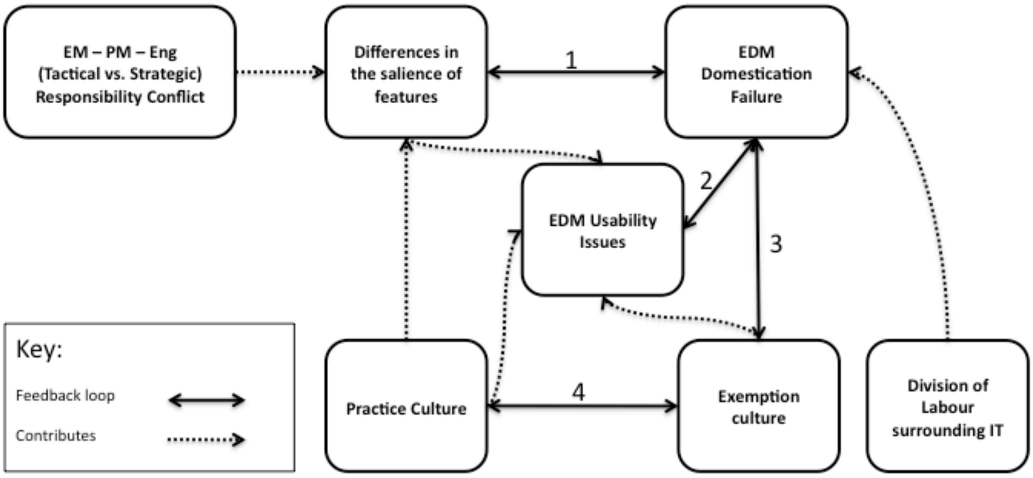


Figure 7.1 - Interactions of the socio-technical issues identified

The first vicious circle occurs between the differences in saliencies of EDM features and the domestication of EDM. The programme managers and engineers within 'Company A' are not valuing the visibility and control features of EDM as much as engineering management. This has resulted in the continued use of shared drives, rather than the use of EDM, resulting in a lack of domestication and thus an absence of familiarity and acceptance of EDM. Conversely as domestication has not occurred, programme managers and engineers will not have had the opportunity to be convinced of the value of the visibility and control features.

The second vicious cycle occurs between domestication and usability issues. Because domestication has not occurred users experience usability issues due lack of familiarity or because of lack of adaptation to the tool over time. Conversely users experience usability issues because of lack of domestication e.g. familiarity / tool adaption.

The third vicious circle occurs between the exemption culture and domestication. The culture of allowing projects to decide on the extent and nature of EDM use (exemption culture) has resulted in a lack of familiarity with the full capabilities of EDM and consequently EDM is not perceived as acceptable as a working drive so shared drives are used as an alternative. Conversely because EDM has not been domesticated this has reinforced the culture of exemptions by giving people a reason not to use the tool.

The fourth vicious circle occurs between the practice culture and the exemption culture. As work is performed on the basis of norms (e.g. individual and shared experience of what has happened in the past) rather than following explicit 'rules' (e.g. referring to process documentation) this has made it acceptable for projects to exempt themselves from standard ways of working such as EDM. Conversely, since projects are permitted to exempt themselves from standard ways of working this reinforces the 'practice culture' as enacting an exemption is in itself the exercise of the primacy of experience over standards and processes.

In summary, we hypothesise that the interaction of these vicious circles contributes to sustaining a situation where the extent and nature of EDM use varies on a project-by-project basis and is dependent upon the preferences of the project teams. In the following sections it will be explained how each of the issues was identified using structural-intentional responsibility modelling.

7.4.3.2 Modelling Intentional Issues

In this subsection we analyse the intentional components of the socio-technical system. We begin by identifying incompatibilities between the human agents' responsibilities, followed by incompatibilities between their objectives and mediators. This analysis focuses on three important agents within the situation:

- Engineering managers, whom are technical subject matter experts and are responsible for improving the overall efficiency, quality, safety and so on of the outputs created by engineers;
- Programme managers, whom are responsible for ensuring the engineers' outputs meet customer expectations and meet contractual obligations;
- Engineers, whom produce system and component designs (and related documentation) under the management of engineering managers and programme managers.

We found that the responsibilities of engineering management, programme management and engineers are aligned such that their overall responsibilities are **positively dependent** (See Figure 7.2[12]). This was deduced from the responsibility diagram by observing the relationships between engineering management's responsibilities and those of the other parties. It may be seen that their responsibility to improve delivering time, quality, efficiency and safety 'helps'

[12] The role of responsibility models in this analysis is to provide a form of visual notation to help the analyst understand the network of relationships between agents' responsibilities.

programme managers and engineers fulfil their responsibilities. The 'helps' relation indicates that improved delivery time and improved quality makes programme management's responsibility of meeting customer expectations and contractual obligations easier to fulfil. It may be observed that this responsibility also 'helps' engineers to meet time, budget and quality pressures, which in turn 'help' programme managers meet customer expectations. The engineers responsibility to follow process also 'helps' engineering managers implement change / process improvement by enabling them to monitor and assess the standardised processes. Overall this means it is in the interest of all parties to coordinate their activities as one party's success 'helps' contribute to the success of the other parties.

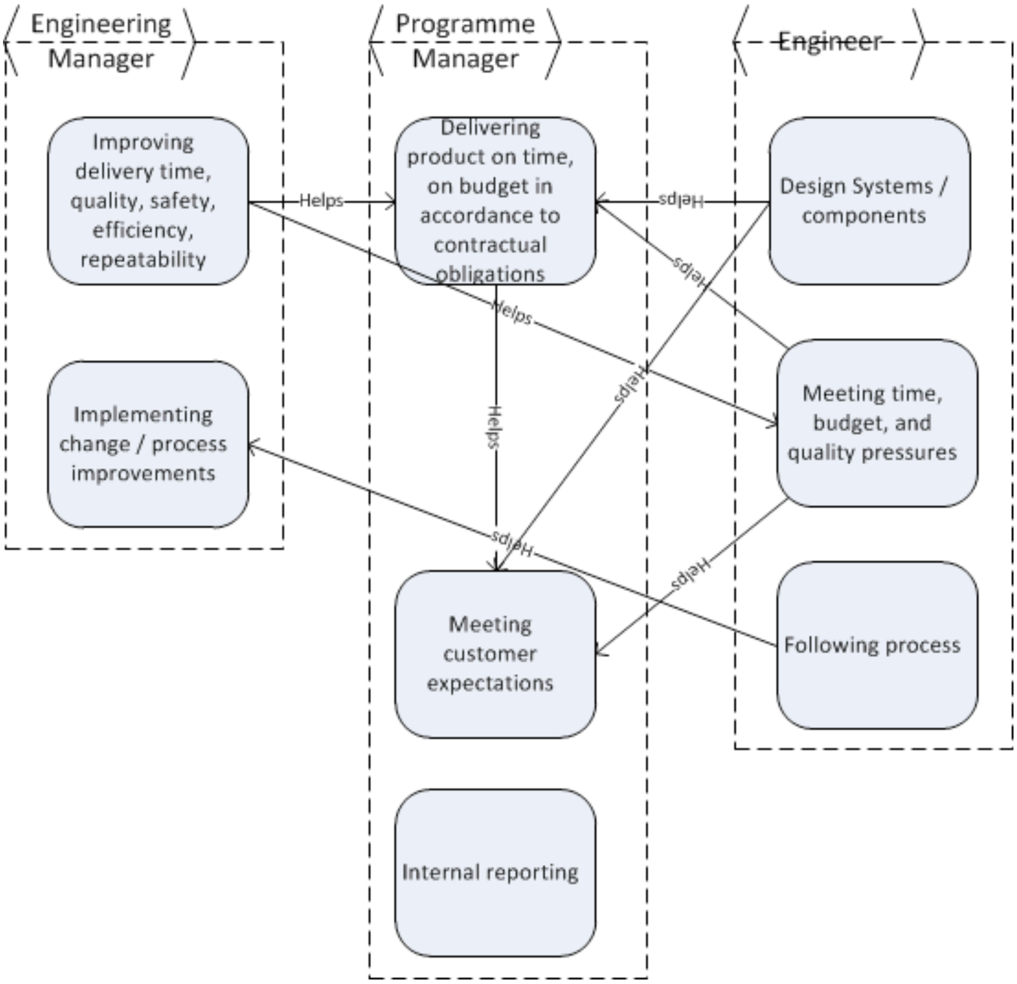

Engineering Manager
Programme Manager
Engineer
Improving delivery time, quality, safety, efficiency, repeatability
Implementing change / process improvements
Delivering product on time, on budget in accordance to contractual obligations
Meeting customer expectations
Internal reporting
Design Systems / components
Meeting time, budget, and quality pressures
Following process
Helps
Helps
Helps
Helps
Helps
Helps
Helps
Helps

Figure 7.2 - Responsibilities Activity Space

Whilst responsibilities are positively dependent, it is possible for process conflict to occur. By process conflict we mean that the *manner* in which parties pursue their responsibilities may be perceived by another to interfere with their own responsibilities. This is investigated in our subsequent analysis of objectives and structural issues.

Analysis of objectives revealed that there is a potential process conflict between the objectives of engineering management and the objectives of programme management (See Figure 7.3). This may be deduced from the diagram by observing the relationships between objectives[13] that are represented as hexagons. One can see that the objective of pursuing improvements via process standardisation and lean thinking potentially interferes with programme manager's objectives of meeting contractual obligations, meeting customer expectations and engineers meeting programme managers expectations. This interference occurs because in the short term the process of implementing changes can caused short-term degradation of performance even if done well. This short-term degradation of performance must be carefully coordinated with programme management to avoid effecting milestones or other legitimate concerns they may hold.

[13] Objectives represent how an agent breaks down a responsibility into a set of objectives to be fulfilled. These objectives will then directly influence the agent's behaviour.

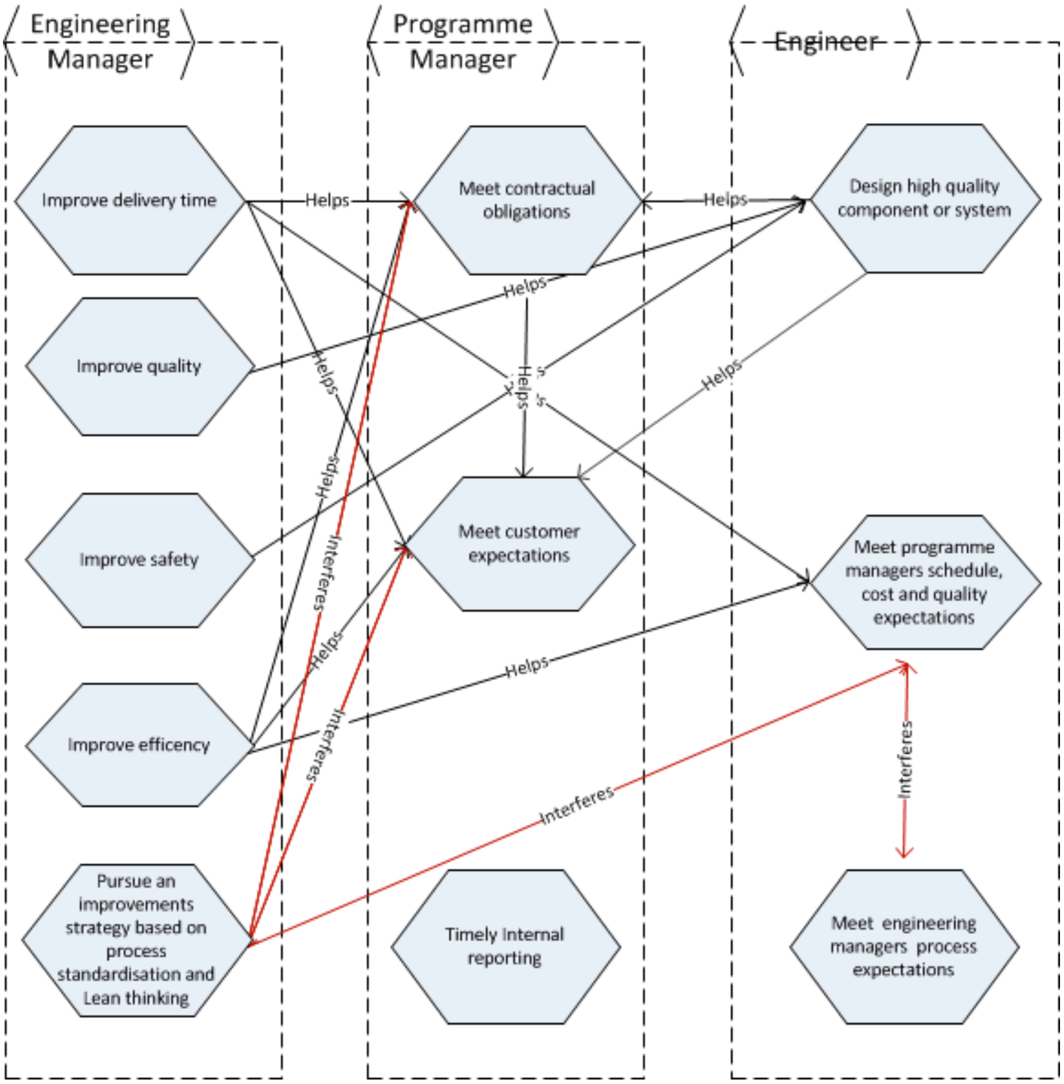

Engineering Manager
Programme Manager
~~Engineer~~
Improve delivery time
Improve quality
Improve safety
Improve efficency
Pursue an improvements strategy based on process standardisation and Lean thinking
Meet contractual obligations
Meet customer expectations
Timely Internal reporting
Design high quality component or system
Meet programme managers schedule, cost and quality expectations
Meet engineering managers process expectations
Helps
Helps
Helps
Helps
Helps
Helps
Helps
Helps
Helps
Interferes
Interferes
Interferes
Interferes

Figure 7.3 - Objectives Activity Space

We now proceed to assessing the mediators of activity in this situation. Mediators are entities that mediate the transformation of responsibilities into objectives and then subsequently into behaviour. In this context the EDM system and its perceived alternative ‘shared drives’ are analysed as they afford agents different ways of fulfilling their responsibilities.

The EDM and ‘shared drives’ were identified as mediators of the tensions that exist between the strategic (long term) and tactical (short term) priorities of ‘Company A’. The EDM provides features such as file access control and audit trails that engineering managers perceive to help them fulfil their responsibilities such as improve safety, product quality and efficiency. The salience of these features to programme managers and engineers is much weaker. Some programme managers and engineers preferring the speed and ‘drag and drop’ ease of ‘shared drives’ over the file access control and audit trail features of the EDM.

This difference in the salience of features, and thus the (lack of) enthusiasm to use the EDM reflects the divide between the long-term priorities of engineering management and the shorter-term priorities of programme managers and engineers. Therefore EDM and 'shared drive' use mediates the division in the strategic (long-term) and tactical (short-term) priorities of 'Company A'.

7.4.3.3 Modelling Structural Issues

In this subsection we analyse the structural components of the socio-technical system. We begin by identifying incompatibilities due to the division of labour, followed by incompatibilities due to community differences and finally incompatibilities due to attitudes towards rule following.

The division of labour with respect to programmes is structured using a matrix approach, such that engineers must meet the requirements of programme managers and those of engineering managers. This means that changes in work activity (e.g. process improvement) should not be inconsistent with either party's interests otherwise resistance to change may be amplified. We believe that inconsistencies have occurred because historically it would appear that 'Company A' has found it challenging to embed standardised ways of working as illustrated by the loss and reacquisition of CMMI level 2. Additionally discussions also suggest that it may be the case that certain programme managers, or product group managers, have reservations about the extent process standardisation and data management practices may contribute to easing their time and budget concerns. This attitude appears to have rippled through to engineers working to time and budget pressures where a culture of de-prioritising the importance of time consuming processes or the use of EDM (a data management tools perceived to be time consuming) seems to be occurring. A strong sign of this culture of de-prioritisation is that the use of EDM was not made mandatory until approximately 8 years after its introduction.

The division of labour can result in disconnects of ownership or responsibility. It appears that this has occurred with respect to the ownership/responsibility associated with the *domestication* of EDM. *Domestication* is the process of making a technology/tool acceptable and familiar to its unique communities. Domestication is not a discrete event (e.g. pre/post-deployment training) but a long-term process that requires the genesis and nurturing of a community of practice and institutional structures to support this community. It does not appear that much domestication has occurred with respect to EDM as end-users are not aware of the existence of a formal user group to liaise with corporate IT to work through technical issues such file upload speeds or screen refresh speeds. Nor are they aware of less formal venues such as forums to discuss best practices. Neither are they proactively informed of 'power-user' tool features that could be relevant to their work. For instance, EDM has features that enable the use of meta-data for improving search accuracy and enabling custom report generation. The quality team use this to generate ISO reports and it is possible that other teams could benefit.

The division of labour can result in different communities that place different saliencies on aspects of their work environment. Such a disconnect has occurred with respect to EDM as engineers, engineering management and programme management have differing and potentially conflicting perceptions of EDM, and the use of shared directories as a viable alternative to EDM. Whilst engineering management value the features that promote visible and controlled working such as version control, audit trails, email notifications of changes to documents, use of aliases to avoid multiple copies, and the ability to control access to documents without the involvement of IT staff. Other communities place different weightings to these aspects or are unaware of their existence. This is evidenced through the extensive use of shared drives where a trade-off for speed and ease of access is made in favour of the visibility and control that EDM offers.

The engineering and programme management communities have a practice rather than process culture. This means that norm following takes precedence over rule following e.g. standards. In other words they perceive that their experiential knowledge is a replacement for adherence to standardised repeatable processes. This is evidenced by:

- discussions revealing that some ‘compulsory’ processes within the quality management system have never been read by anyone other than the authors and reviewers of the processes.
- discussions that suggest that projects are given sufficient flexibility to adopt, adapt or exempt themselves from company standards with relative ease and without having to report this to process improvements for review. This reinforces a culture of practice over process and also results in potential process weaknesses not being reported or corrected.
- the use of EDM project area was not made compulsory until 2009.

7.4.3.4 Modelling the Outcome: The Product of the Cognitive & Structural Issues

The resultant outcome of the interactions between cognitive and structural issues identified in the preceding sections is presented in Figure 7.4. The extent and nature of EDM use varies on a project-by-project basis and is dependent upon individual programme managers and engineering teams. Figure 7.4 shows that engineering management are continuing to encourage EDM use and are continuing to pursue an improvement strategy based on the concept of standardisation.

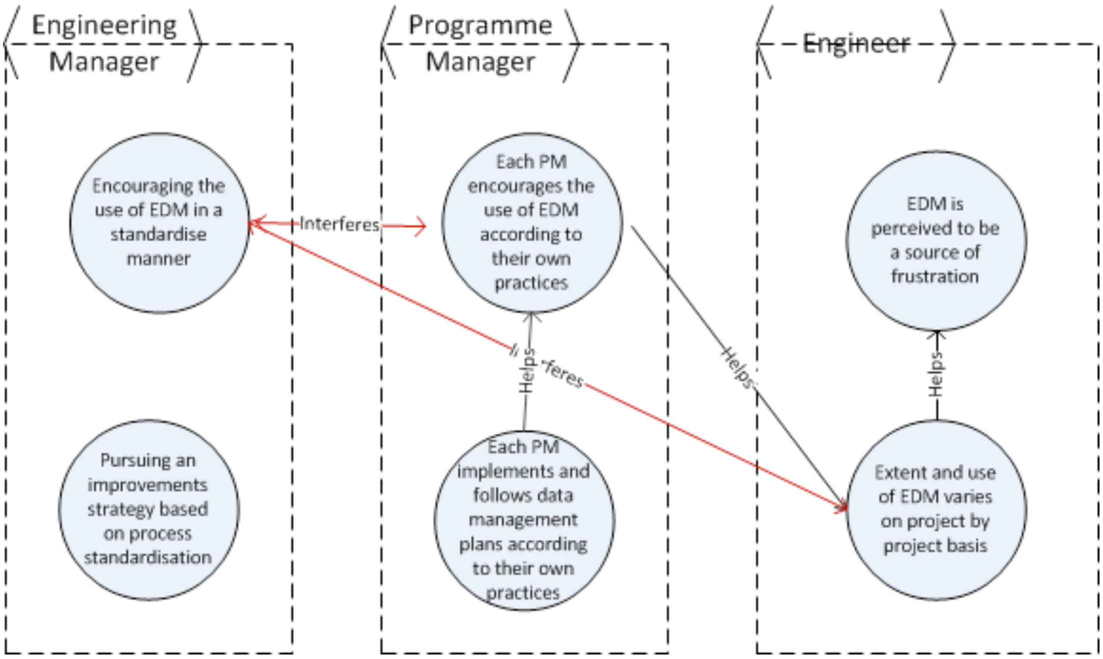

Engineering Manager
Programme Manager
Engineer
Encouraging the use of EDM in a standardise manner
Interferes
Each PM encourages the use of EDM according to their own practices
EDM is perceived to be a source of frustration
Helps
Interferes
Helps
Helps
Pursuing an improvements strategy based on process standardisation
Each PM implements and follows data management plans according to their own practices
Extent and use of EDM varies on project by project basis

Figure 7.4 - Outcome Activity Space

The outcome may be summarised as the product of the interactions of the following socio-technical components:

Division of labour

- A matrix structure is a conduit of tensions, or potential conflicts, and therefore employees within the matrix become a focal point of resistance when incompatible demands are placed.
- The division of labour can result disconnects of responsibility or ownership. This has occurred with respect to the *domestication* of EDM.

Communities

- Communities within ‘Company A’ have emerged around the division of labour and thus roles/responsibilities e.g. program management, engineering management, engineers. There is a potentially conflicting divergence between the viewpoints of each of these communities with respect to the value of EDM and the salience of its capabilities and purpose.

Rules

- There is a strong practice culture rather than a process culture within programme management and engineering. This means that work is performed on the basis of norms (e.g. individual and shared experience of what has happened in the past) rather than following explicit ‘rules’ (e.g. referring to process documentation).

Responsibilities

- The responsibilities of engineering management, programme management and engineers are aligned such that their overall responsibilities are **positively dependent**. This means it is in all parties' interests to coordinate their activities as one party's success contributes to the success of the other parties.
- Whilst responsibilities are positively dependent. There is however scope for process conflict and our study suggests that may be occurring. By process conflict we mean that the manner in which parties are pursuing their objectives may be perceived by others as interfering with their own objectives.

Objectives

- There is a potential for conflict between the objectives of engineering management and programme management. Whilst it is the responsibility of engineering management to take a strategic (long-term) view and run improvement projects. This is potentially conflicting with the tactical (shorter-term) responsibilities of programme management. Introducing change, even when successful, can cause short-term productivity degradation as changes are being 'bedded in'. This can be at odds with programme managers' contractual obligations such as milestones.
- Note: This potential for conflict between engineering management and programme management ripples through to engineers working on projects by virtue of the matrix organisational structure.
- Note: It is possible that the length of the 'bedding in' process is protracted by parties that allow temporary exemption from the change being 'bedded in'. This may perhaps cause a **vicious circle** where changes are never fully adopted due to culture of exemption.

Mediators

- EDM as a mediator of work is acknowledged by all communities to suffer from usability issues at both the software and system level, which ultimately results in frustration.
- The salience of these usability issues as a reason to use shared drives differs between engineering management, programme management and engineers.

7.5 Discussion

Our study was designed to evaluate the following hypothesis: 'structural-intentional responsibility analysis enables the identification of stakeholder conflicts in enterprise-scale deployments by identifying incompatibilities between structural and intentional elements'. Our study supports this hypothesis as firstly we identified a number of tensions between structural and intentional elements and their interactions. Secondly we were able to make specific recommendations to ameliorate the deployment.

7.5.1 Recommendations

The recommendations we made comprised a six-step plan to address the vicious circles identified. Each individual recommendation can be described as codified commonsense. The value of the plan came from the fact that the *combination* of recommendations we made was tailored to the specific dynamics of the deployment environment.

We proposed that our recommendations could be implemented in the following manner:

1. -Identify a minimal set of practices for EDM use that programme management and engineering management have firm grounds to believe are valuable to Company A (taking into account short term vs. long term tradeoffs).
   -Assign ownership of domestication to a high level manager(s) within Company A
2. -Create a network of EDM evangelists with members from each discipline / functional unit
   -Create a EDM steering committee with members representing the interests of programme management, engineering management, IT and other legitimate stakeholders.
3. -Support the EDM evangelists to create role specific resources that can be used to facilitate the use of EDM according to the set of practices intended by management
   -Actively inform end-user of EDM steering committee and its role in improving the tool
4. -Actively inform end-users of the strategic value of EDM and the mandatory practices
   -Actively inform end-users of the role specific feature of EDM and how the mandatory practices will change there existing work practices
5. -Roll out mandatory practices
6. -Audit practices in use to ensure they are in accordance with mandatory practices

These recommendations were derived from identifying approaches for dealing with each of the issues identified by structural-intentional responsibility analysis. For instance to break the culture of programmes exempting themselves from EDM usage, the use of EDM should be made mandatory for specific practices that programme management and engineering management have identified, on a firm basis, as valuable to Company A. These practices should be clearly prescribed, perhaps as a set of programme data management practices, and audited for compliance to signal that exemption from these is not a cultural norm. We believe that the implementation of this change should be coordinated with the programme management community in order to minimise the potential for conflict due to possible tactical/strategic differences in priorities.

To address the domestication issue we suggested that management should take on ownership of the responsibility for EDM's domestication and in doing so create appropriate institutional structures and resources to support its occurrence.

1. Staff should be *actively informed* of the value-adding features of EDM for their *specific* work activity.

Staff should be shown what previous practices/tools EDM replaces and what new practices/responsibilities it creates and how it contributes value to Company A. This should not be mistaken for merely informing user of the system's functionality. Staff should also be informed on an ongoing basis rather than as one off training to reinforce the importance of the systems use. Sending email reminders, creating posters, distributing 'cheat sheets' are all appropriate means of enacting an ongoing basis.

2. Institutional structures should be created to enable the coordinated *adaptation* of EDM.

A formal structure should be in place for staff to raise usability issues and have the case considered for a possible adaptation to EDM. The creation of an EDM steering group that represents the interests of EDM users and other stakeholders across the site could enable this. A typical steering group would have representatives from a broad range of functional units including users, managers and IT representatives, and would have the authority to make change requests with respect to EDM on behalf of the site.

3. Institutional structures should be created to nurture/support an EDM community of practice so that best practices and advice can be shared.

Specially selected members of each discipline, or functional unit, should be empowered by giving them ownership of the task of evangelising their local community to follow EDM best practice. The EDM evangelist's role would be to a) share/nurture/support best practice and b) learn from the experiences of their colleagues from across disciplines/functional units. This network of EDM evangelists could be supported by the creation a homepage/wiki containing resources to support their task by providing role specific 'cheat sheets', how-to guides, and best practice guidelines.

To address the issue of salience of features effort should be put into fostering a shared view of tool features. To achieve this we suggested that staff should be actively informed, or reminded, of the strategic organisational need for capabilities that EDM offers. This can be implemented by top down managerial communication and reinforced by a network of local EDM evangelists.

To address usability issues we suggested that an institutional structure should be created to enable the adaptation of EDM and the folder structure. The usability issues that we highlight could be considered by an EDM steering committee created to address issues of domestication.

To address the practice culture we suggested that exemptions from standard practices should be very carefully controlled to ensure that: i) they are necessary; ii) process improvement can take place to ensure that over time exemptions become the exception rather than the norm. This can be implemented by auditing the use of EDM, enforcing standard practice where appropriate, and highlighting opportunities for process improvement to appropriate parties e.g. process improvements or EDM steering group.

7.5.2 Reflections on the Use of Structural-Intentional Responsibility Modelling

Structural-intentional responsibility models provided an appropriate means of understanding the relationships between responsibilities, objectives and outcomes. Responsibility models provided an important means of concisely representing the insights gathered using issue maps of interview transcriptions. Their most important contribution is to bring to the foreground the *relationships* between issues so that their interconnections may be understood. A limitation of this approach was however observed. When the number of entities (responsibilities, objectives, outcomes) or relations became large the diagrams became cluttered and were no longer as helpful in articulating the situation. This limitation is addressed in the next chapter via means of representing the models as a directed graph and using a large-scale network analysis and visualisation package (Gephi.org) to represent the graph in a user-friendly manner.

In comparison to a task centred analysis approaches (such as I*) we believe that, despite the above-described limitations, structural-intentional responsibility analysis offers a number of useful tradeoffs. Firstly data collection can be of a shorter duration as a detailed understanding of tasks is not required and thus avoids time-consuming ethnography or process mapping. Secondly, the scale of the deployment under analysis can be much larger as data collection is rapid and data analysis can be supported through the use of off-the-shelf digraph visualisation and analysis tools that support large datasets. It remains an open research question if task centric approaches such as I* models can scale up to analyse large-scale systems (E. Yu et al., 2011).

The disadvantages of a structural-intentional approach, in contrast to a task centric approach, is that it will not deliver insights with respect to the subtleties of task level interactions within a work environment e.g. distributed coordination, awareness, spatial and temporal organisation and so on. Nor does structural-intentional analysis enable modelling at the task level so it does not represent how actors and resources are configured at a task level.

Despite these shortcomings, we believe that this trade-off is desirable as it makes structural-intentional analysis complementary to established task/activity centric analyses that inform information systems development. For example when time permits ethnography, we expect structural-intentional analysis to provide complementary findings.

7.6 Conclusion

This work illustrates that structural-intentional responsibility modelling is a viable candidate as a scalable engineering technique for analysing and troubleshooting enterprise-systems post-implementation. Our case study indicates that structural-intentional analysis has a number of attractive characteristics with respect to timeliness and scalability. Data collection appears more rapid than either process mapping or ethnography and data analysis appears extremely scalable as it can be supported through the use of off-the-shelf digraph visualisation and analysis tools. Our fieldwork demonstrates that structural-intentional data is: i) sufficient to diagnose problematic interactions between a system, intentional elements and structural elements; ii) sufficient to suggest practical interventions to ameliorate a deployment. We therefore advocate the structural-intentional approach as a candidate engineering approach for analysing and troubleshooting large-scale deployments.

Our conclusion is limited by the usual limitations of qualitative case study research. Case study research may not be generalisable and whilst every effort was taken to minimise investigator or participant bias, bias may be reflected in our findings.

There are many opportunities to further validate and develop the structural-intentional view of deployments. We encourage more case studies or action research to demonstrate its scalability and ability to ameliorate deployments in a variety of settings. We encourage comparative work between structural-intentional approaches and task/activity-centric approaches (such as I*) to explore their strengths and weaknesses. We also encourage the development of tools to support structural-intentional analysis.

8. Scaling Structural-Intentional Responsibility Modelling Using Network Analysis

8.1 Introduction

In recent years developments in the analysis of complex networks have enabled scientists to analyse certain classes of large-scale socio-technical systems (Vespignani, 2012). Complex network analysis has enabled physicists, chemists, epidemiologists and social scientists to study systems comprising thousands or millions of nodes in applications as wide ranging as social network analysis and metabolic pathway analysis. This chapter proposes that these techniques may also be of practical use in the analysis of problematic socio-technical systems in organisational settings.

Troubleshooting large-scale systems remains an open research question within the socio-technical systems engineering community. Whilst analysing socio-technical issues in small-to-medium scale systems may be achievable using approaches such as theoretically informed rapid ethnography, analysing large scale systems is an open research question because of the *overwhelming* number of socio-technical elements and interactions involved (RAE, 2004).

In order to develop a more scalable approach, it was argued in chapter 7 that the exploration of structural and intentional factors using interviews may enable a more scalable way of analysing a system as it has a different set of trade offs (Greenwood & Sommerville, 2011a) to typical ethnographic approaches. The structural-intentional approach is similar in spirit to SSM but incorporates a variant of the 'activity space' framework (Halloran, 2000, 2001) - a framework that analyses a broader range of socio-technical elements than SSM.

This chapter provides evidence that problematic large-scale systems are amenable to computer-aided analysis using network theory. This is significant to the problem of scalability as it enables the partial automation of the process of identifying important elements and complex behaviour thus reducing the burden of work and perhaps enabling the analysis of large systems where a human cannot be expected to analyse every element in the problem due to their vast number.

This chapter demonstrates, using the EDM case study presented in chapter 7, proof-of-concept tools for large-scale network analysis and visualisation that may provide a promising avenue for identifying problematic elements and interactions amongst an overwhelming number of socio-technical elements. We demonstrate the potential of this approach by showing that:

i) a system may be represented as a directed graph such that the elements in the system are represented as nodes, and interactions between nodes as links;

ii) that eigenvector centrality, a well established measure of node importance, may be used to rank the importance of elements in a system and that highly ranked elements match those identified as important by a human analyst;

iii) the 'complexity' of a system, or a part of a system, may be characterised using a feedback degree score which provides an indication of the extent elements are highly interconnected and are involved in feedback loops. The implications of these findings are that computers may be used to aid the analysis of problematic large-scale socio-technical systems by highlighting elements, or groups of interacting elements, that are important to the overall outcome of a problematic system. This contribution is significant as it provides an avenue for developing scalable engineering techniques to troubleshoot large-scale systems.

This chapter is structured such that section 2 provides an overview of aspects of complex network analysis that may be relevant to socio-technical analysis. In section 3 we describe our aim, hypotheses and research design. In section 4 we illustrate our approach to representing, visualising and analysing a problematic socio-technical system as a directed graph. In section 5 we present our case study findings and analyse the relationship between: i) eigenvector centrality scores and node importance as ranked by an analyst; ii) feedback degree scores and node complexity as ranked by an analyst. Finally in section 6 we conclude and describe future research opportunities.

8.2 Using Network Analysis to Analyse Socio-Technical Systems?

Network theory is a rapidly evolving branch of computer science and mathematics that studies mathematical structures that model relations between discrete objects. Complex networks are "networks whose structure is irregular, complex and dynamically evolving overtime" (Boccaletti et al., 2006). In recent years network analysis and visualisation tools have become accessible to non-mathematicians due to the increasingly widespread availability of off-the-shelf tools. These tools have enabled the analysis of systems of thousands or millions of nodes such as electric power grids, the Internet and social networks. This ability to analyse vast numbers of nodes and interactions makes network analysis attractive for the purpose of developing scalable techniques for analysing problematic large-scale systems.

There are two significant barriers to using network analysis for the analysis of problematic socio-technical systems. Firstly the socio-technical system must be representable as a set of nodes and a set of links. Secondly, appropriate network analysis techniques or metrics must be identified, or invented, to enable the identification of nodes or links that are important to sustaining the overall problematic behaviour of a system.

In this chapter we propose that a socio-technical system is representable as a directed graph using the elements and relationships used in structural-intentional responsibility modelling, as described in chapter 7. That is to say nodes represent agents' responsibilities, objectives, outcomes, resources, the division of labour and attitudes to rule following. The links between nodes represent 'helps' and 'interferes' relations as per chapter 7. The directed graph therefore is a representation of the structural-intentional elements composing the system and the tensions between them.

To address the barrier of appropriate techniques / metrics, in this chapter we explore the usefulness of well-established metrics. Firstly, we test the usefulness of network centrality metrics to discover the 'influence' or 'importance' of a structural-intentional element (node) in a socio-technical system (directed graph). Network centrality metrics have been used for this purpose in other domains such a social network analysis (Boccaletti et al., 2006). Secondly we explore whether the 'complexity' (interconnectedness and extent of feedback) of an element may be estimable by combining existing techniques to count the number of feedback loops a node is involved in and its number of links to other nodes. The counting of loops, referred to as *k*-cycles, is a well-established practice (Vázquez et al., 2005) as is counting the links of node (Opsahl et al., 2010).

A particularly well-suited centrality metric for our purposes is *eigenvector centrality*. The metric captures the intuition that a node that influences many other nodes is influential, whilst also taking into account the notion that a node that influences many highly influential nodes is more influential than a node that influences many weakly influential nodes. A node's eigenvector centrality (Bonacich, 1972) is defined as its summed connections to others weighted by their centralities. The centrality of node i is given by $\lambda \upsilon_i = \Sigma A_{ij} \upsilon_j$ - the defining equation in matrix notation is $\lambda \upsilon = \mathbf{A} \upsilon$ where $\mathbf{A}$ is the adjacency matrix, λ is a constant and υ is the eigenvector. The metric is particularly well-suited for our purpose as it has the following properties: it assumes that all nodes influence their neighbours (rather than influence being restricted to a single shortest path) (Borgatti, 2005); influence is subject to feedback loops such that nodes are revisited multiple times (rather than assuming no node is visited more than once) (Borgatti, 2005).

We propose that the 'complexity' (extent of interconnectedness and of being involved in feedback loops) of a node may be captured using a metric that we call feedback degree. It captures the intuition that the complexity of a node is primarily determined by the complicatedness of its interactions (number of feedback loops it partakes) and secondarily by its total number of interactions. Mathematically it comprises the linear combination of a nodes *degree* and the number of *k-cycles* (feedback loops) it partakes. The degree of a node in a directed graph is the sum of the number of outgoing links (termed outdegree) and incoming links (termed indegree). The defining equation for the outdegree is $k_i^{out} = \Sigma_j a_{ij}$, the indegree is $k_i^{in} = \Sigma_j a_{ji}$, and the total degree is $k_i = k_i^{out} + k_i^{in}$, where k_i is the degree for the node i and a_{ij} represents an element in the adjacency matrix of the network. We define feedback degree as $f_i = k_i/2n + c_i$ where f_i is the feedback degree for a node i, k_i is the degree of the node i, n is the total number of nodes in the network, and c_i is the number of cycles the node partakes.

8.3 Research Design

This research analysed data obtained from the case study in chapter 7. Our aim was to provide proof of concept that techniques for network analysis help to analyse real world problematic socio-technical systems. We formulated the following hypotheses from our aim:

1. Eigenvector centrality may be used to rank the importance of elements in a system i.e. high eigenvector centrality scoring nodes correspond to those identified as 'most important' by a human analyst. Our null hypothesis was that there would be no difference between the distributions of eigenvector centrality scores for the population of important elements and unimportant elements.
2. Feedback degree may be used to rank the 'complexity' of elements in a system, or a part of a system i.e. feedback degree scores correlate with node complexity as ranked by a human analyst. Our null hypothesis was that any correlation between node complexity as ranked by a human analyst and feedback degree would not be statistically significant.

8.3.1 Data Collection and Analysis

To test the hypotheses outlined above we firstly analysed the elements and interactions (without the aid of network analysis software) to identify important elements and interactions that mediate the problematic outcome. This process is described in chapter 7. Secondly we repeated the analysis with the aid of network analysis and visualisation software (Gephi.org) by parsing the elements and interactions into a machine-readable file format (DOT language).

To assess hypothesis 1 we divided the nodes into two populations, those thought 'most important' by the analyst and the rest that we labelled 'least important'. Secondly we generated eigenvector centrality scores for each node using Gephi. Thirdly, we used an Independent-Samples Man-Whitney U-test to compare the distribution of eigenvector centrality score between each population using the PASW (SPSS) statistical package. Our null hypothesis for hypothesis 1 was that there would be no difference between the distributions of the 'most important' and 'least important' populations thus confirming that eigenvector centrality is not an indicator of importance. To confirm hypothesis 1 we expected each population to have a different distribution and that the 'most important' population would have a larger median and mean eigenvector centrality score than the 'least important' population. This would enable us to conclude that elements with a large eigenvector centrality are more likely to be members of the 'most important' population than the 'least important' population; thus confirming that ranking elements by eigenvector centrality is a reasonable method of ranking the importance of an element.

To assess hypothesis 2 we generated three sub graphs of the problematic system. We used sub graphs comprising up-to 10 nodes, rather than the whole graph comprising 30+ nodes, to make the ranking scenario a manageable size for a human analyst. Subsequently:

1. We (without the aid of network analysis software) ranked each element on the basis of its complexity as judged by a human analyst;
2. We generated feedback degree scores for each node;
3. We used Spearman's correlation to analyse the relationship between feedback degree and a node's complexity as judged by an analyst. Spearman's correlation is a non-parametric measure of statistical dependence between two variables and is also implemented in PASW.

Our null hypothesis was that any correlation between feedback degree and complexity would be due to random chance. To confirm hypothesis 2 we firstly expected a statistically significant correlation between feedback degree and complexity. Secondly we desired the $r>0.83$ such that feedback degree accounts for a meaningful proportion of the variance (>70%) of the complexity as judged by an analyst. This would enable us to conclude that feedback degree is a reasonable indicator of the complexity of an element in a problematic system.

8.4 Representing a Responsibility Model as a Direct Graph

We represented the elements and interactions composing our problematic system using a plain text graph description language called DOT (Gansner & North, 2000). The language enables the description of a network in a machine-readable format so that a network analysis and visualisation package, such as Gephi (Bastian, Heymann, & Jacomy, 2009), can process the data. The DOT language has a simple syntax that enables the expression of nodes, links and presentational information such as shapes, colours and labels.

The language offers a small yet expressive set of primitives that enable the description of a graph with a finite number of nodes and links. The key primitives that we made use of included:

digraph g { … } – which defines a directed graph called ‘g’ and the description of the graph in contained within the parentheses;

[colour = …] – which sets the colour attribute of a node, or a link, to a specified value

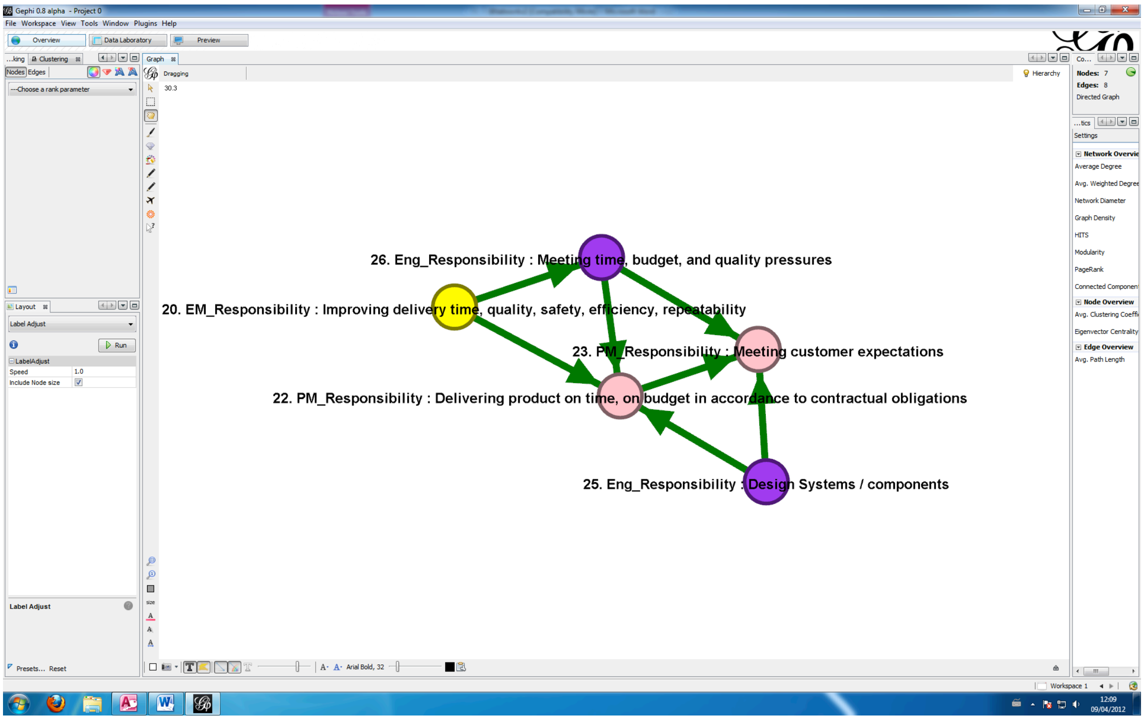
[label = …] – which sets the label attribute of a node, or a link, to a specified value

-> - which is an edge operator that connects one node to another e.g. “a” -> “b”;

A directed graph (See Figure 8.1) representing the interactions between the responsibilities in the problematic system described in section 5.1 (See Figure 8.2) is represented in DOT language below:

```
digraph g {
"20. EM_Responsibility : Improving delivery time, quality, safety,
efficiency, repeatability"[color=yellow];
"22. PM_Responsibility : Delivering product on time, on budget in
accordance to contractual obligations"[color=pink];
"20. EM_Responsibility : Improving delivery time, quality, safety,
efficiency, repeatability" -> "22. PM_Responsibility : Delivering product
on time, on budget in accordance to contractual obligations"[color=green,
label = "" ] ;
"21. EM_Responsibility : Implementing change / process
improvements"[color=yellow];
"26. Eng_Responsibility : Meeting time, budget, and quality
pressures"[color=purple];
"21. EM_Responsibility : Implementing change / process improvements" ->
"26. Eng_Responsibility : Meeting time, budget, and quality
pressures"[color=green, label = "" ] ;
"22. PM_Responsibility : Delivering product on time, on budget in
accordance to contractual obligations"[color=pink];
"23. PM_Responsibility : Meeting customer expectations"[color=pink];
"22. PM_Responsibility : Delivering product on time, on budget in
accordance to contractual obligations" -> "23. PM_Responsibility :
Meeting customer expectations"[color=green, label = "" ] ;
"25. Eng_Responsibility : Design Systems / components"[color=purple];
"23. PM_Responsibility : Meeting customer expectations"[color=pink];
"25. Eng_Responsibility : Design Systems / components" -> "23.
PM_Responsibility : Meeting customer expectations"[color=green, label =
"" ] ;
"26. Eng_Responsibility : Meeting time, budget, and quality
pressures"[color=purple];
"23. PM_Responsibility : Meeting customer expectations"[color=pink];
"26. Eng_Responsibility : Meeting time, budget, and quality pressures" ->
"23. PM_Responsibility : Meeting customer expectations"[color=green,
label = "" ] ;
"27. Eng_Responsibility : Following process"[color=purple];
"21. EM_Responsibility : Implementing change / process
improvements"[color=yellow];
"27. Eng_Responsibility : Following process" -> "21. EM_Responsibility :
Implementing change / process improvements"[color=green, label = "" ] ;

}
```



Gephi 0.8 alpha - Project 0
File Workspace View Tools Window Plugins Help
Overview
Data Laboratory
Preview
Clustering
Nodes Edges
---Choose a rank parameter
Graph
Dragging
Hierarchy
26. Eng_Responsibility : Meeting time, budget, and quality pressures
20. EM_Responsibility : Improving delivery time, quality, safety, efficiency, repeatability
23. PM_Responsibility : Meeting customer expectations
22. PM_Responsibility : Delivering product on time, on budget in accordance to contractual obligations
25. Eng_Responsibility : Design Systems / components
Layout
Label Adjust
Run
Label Adjust
Presets... Reset
Settings
Network Overview
Average Degree
Avg. Weighted Degree
Network Diameter
Graph Density
HITS
Modularity
PageRank
Connected Components
Node Overview
Avg. Clustering Coefficient
Eigenvector Centrality
Edge Overview
Avg. Path Length
Arial Bold, 32
Workspace 1
12:09
09/04/2012

Nodes: 7
Edges: 8
Directed Graph

LabelAdjust
Speed 1.0
Include Node size ☑

Figure 8.1 – Sub graph of a problematic system being visualised using Gephi

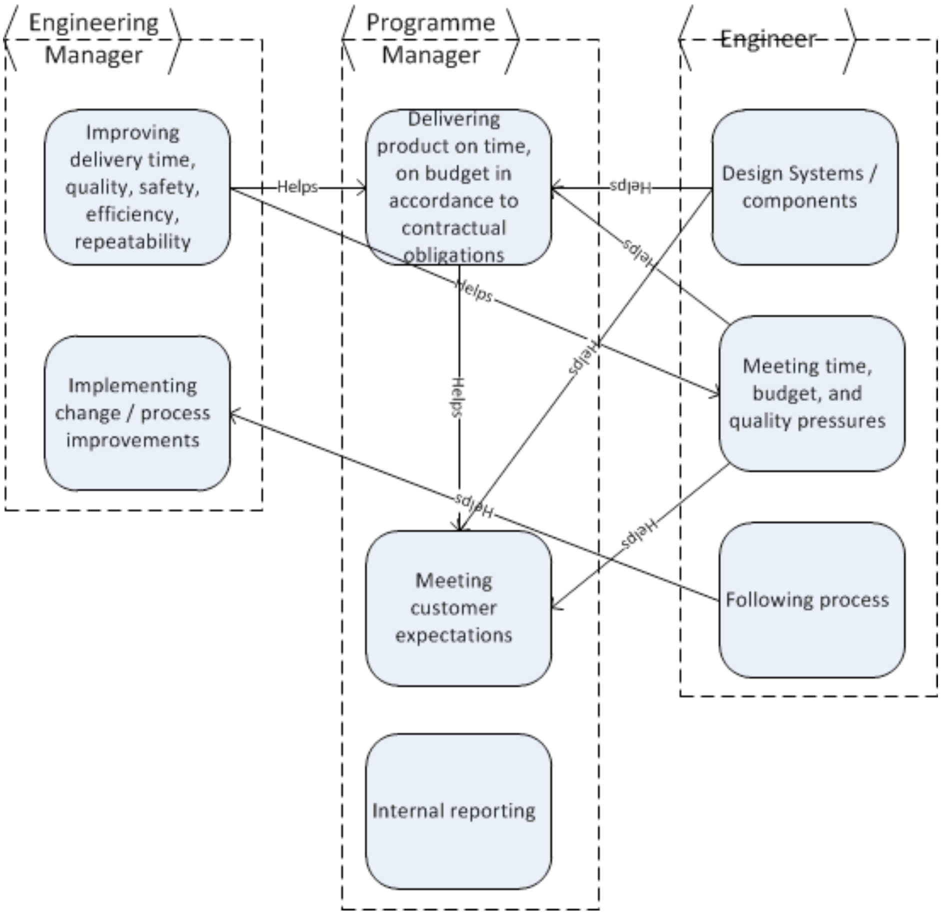
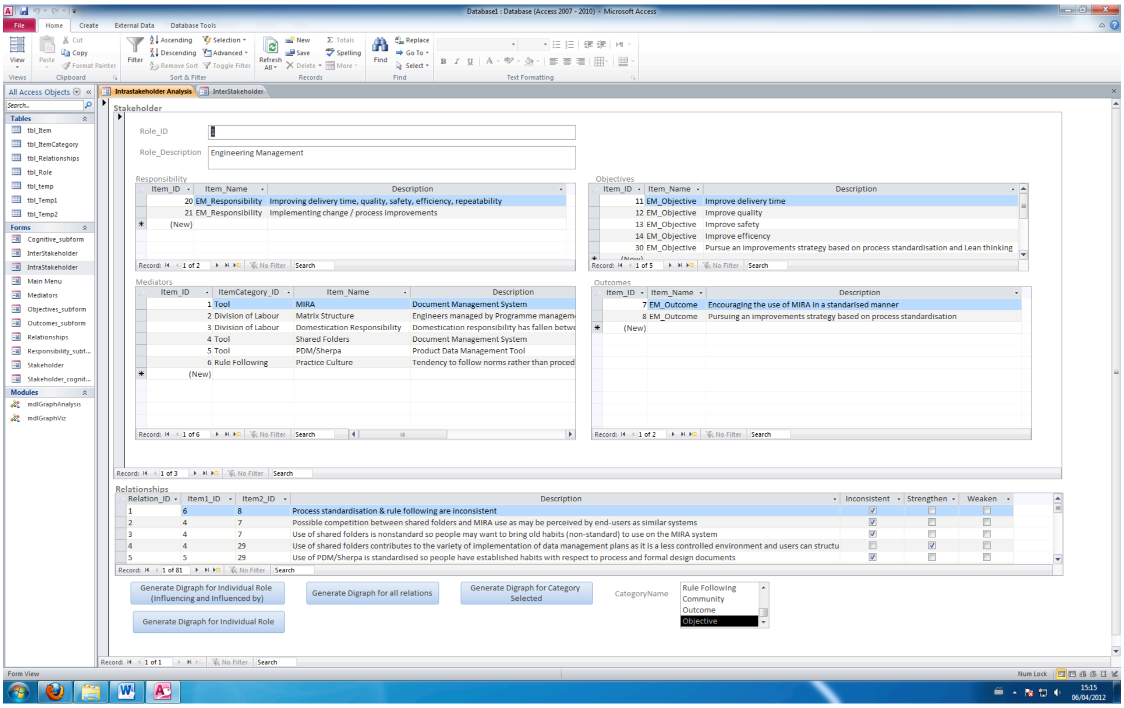

Engineering Manager
Programme Manager
~~Engineer~~
Improving delivery time, quality, safety, efficiency, repeatability
Implementing change / process improvements
Delivering product on time, on budget in accordance to contractual obligations
Meeting customer expectations
Internal reporting
Design Systems / components
Meeting time, budget, and quality pressures
Following process
Helps
Helps
Helps
Helps
Helps
Helps
Helps
Helps

Figure 8.2 - Responsibility Model from Chapter 7

On comparing Figure 8.1 and 8.2 one may observe that they are representations of the same information. For instance, the engineering manager's responsibility for improving delivery time, quality, safety, efficiency and repeatability helps both the programme manager's responsibility to delivery product on time, on budget in accordance to contractual obligations and helps engineer's meet time and budget pressures.

A useful aspect of the language for the purpose of representing problematic systems is its ability to express the colour of nodes and relationships. We have found this feature useful for highlighting node and relationship types when visualising systems with many nodes. In Figure 8.1 the yellow nodes belong to engineering managers, the purple to engineers and the pink to programme managers. The colour of the links between the nodes represents the type of relationship. A green link represents one node supporting another. An orange/brown link represents a tension (potential incompatibility). Visualisation is a particularly powerful technique as seeing the relationships laid out spatially with colours identifying each node type enables the complexity of a problematic system to be articulated extremely rapidly.

Since writing and rewriting DOT files by hand can be tedious and time consuming we developed a prototype GUI using Microsoft Access 2010.



Stakeholder

Role_ID: 1

Role_Description: Engineering Management

Responsibility

Item_ID	Item_Name	Description
20	EM_Responsibility	Improving delivery time, quality, safety, efficiency, repeatability
21	EM_Responsibility	Implementing change / process improvements
(New)		

Record: 1 of 2 No Filter Search

Objectives

Item_ID	Item_Name	Description
11	EM_Objective	Improve delivery time
12	EM_Objective	Improve quality
13	EM_Objective	Improve safety
14	EM_Objective	Improve efficency
30	EM_Objective	Pursue an improvements strategy based on process standardisation and Lean thinking
(New)		

Record: 1 of 5 No Filter Search

Mediators

Item_ID	ItemCategory_ID	Item_Name	Description
1	Tool	MIRA	Document Management System
2	Division of Labour	Matrix Structure	Engineers managed by Programme managem
3	Division of Labour	Domestication Responsibility	Domestication responsibility has fallen betwe
4	Tool	Shared Folders	Document Management System
5	Tool	PDM/Sherpa	Product Data Management Tool
6	Rule Following	Practice Culture	Tendency to follow norms rather than proced
(New)			

Record: 1 of 6 No Filter Search

Outcomes

Item_ID	Item_Name	Description
7	EM_Outcome	Encouraging the use of MIRA in a standarised manner
8	EM_Outcome	Pursuing an improvements strategy based on process standardisation
(New)		

Record: 1 of 2 No Filter Search

Record: 1 of 3 No Filter Search

Relationships

Relation_ID	Item1_ID	Item2_ID	Description	Inconsistent	Strengthen	Weaken
1	6	8	Process standardisation & rule following are inconsistent	☑	☐	☐
2	4	7	Possible competition between shared folders and MIRA use as may be perceived by end-users as similar systems	☑	☐	☐
3	4	7	Use of shared folders is nonstandard so people may want to bring old habits (non-standard) to use on the MIRA system	☑	☐	☐
4	4	29	Use of shared folders contributes to the variety of implementation of data management plans as it is a less controlled environment and users can structu	☐	☑	☐
5	5	29	Use of PDM/Sherpa is standardised so people have established habits with respect to process and formal design documents	☑	☐	☐

Record: 1 of 81 No Filter Search

CategoryName: Rule Following / Community / Outcome / Objective (selected: Objective)

Record: 1 of 1 No Filter Search

Figure 8.3 - Screenshot of Microsoft Access Based GUI

This enables the analyst to rapidly enter each socio-technical element and subsequently indicate which elements interact together using a spreadsheet like interface rather than via code. This enables graphs to be generated, viewed, reviewed or modified rapidly using features typically available in any database or spreadsheet e.g. filters, queries and so on. We found this to be very fruitful when combined with Gephi as it enabled us to rapidly visualise a graph or perform statistical analyses.

8.5 Research Findings

In this section we compare the results of our unaided analysis in chapter 7 with our network aided analysis. By comparing our findings using statistical techniques we analyse:

1. the relationship between important nodes identified by an analyst and a node's eigenvector centrality ranking;
2. the relationship between the complexity of nodes as ranked by an analyst and a node's feedback degree ranking.

8.5.1 Findings from Network Aided Analysis

To test whether eigenvector centrality is a good indicator of element importance we firstly categorised the nodes into two groups – those that we believed to be 'most important' (See Table 8.1) and the rest we labelled 'least important'. The 'most important nodes' were those that we believe contributed most to creating and sustaining the problematic system according to our unaided analysis. The diagram below (Figure 8.4) provides a reminder of these findings from chapter 7, and thus what elements were considered to be 'most important' by the analyst.

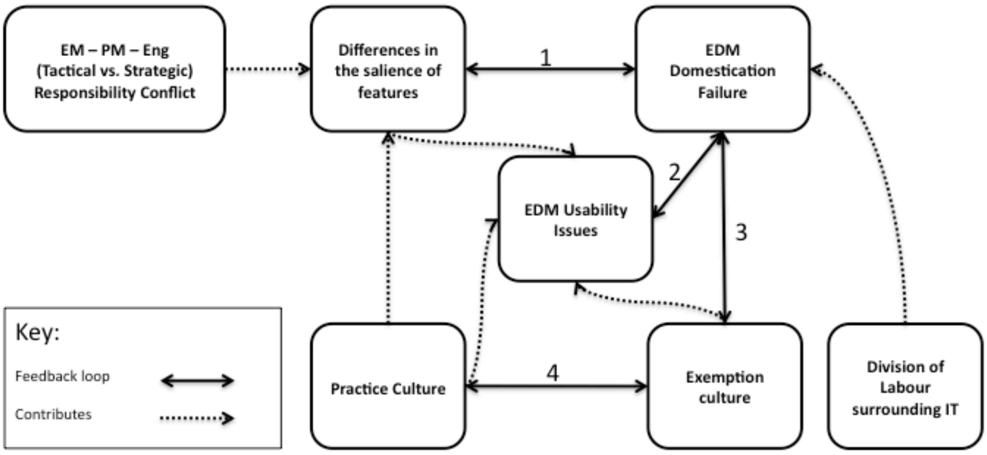


Figure 8.4 – Socio-Technical Interactions Identified in Chapter 7

The reader may observe that the elements in Table 8.1 are those that the analyst perceived to contribute most to the vicious circles. For instance, node '3 Domestication Responsibility' supports the domestication failure element and thus is present in Table 8.1. Node '6 Practice Culture' supports the practice culture element and therefore is present in Table 8.1. The same may be said of nodes '7 EM_Outcome' and '8 EM_Outcome', which sustain the EM-PM responsibility conflict. Node '9 PM_Outcome' supports both the practice culture and exemption culture elements represented. Node '10 Eng-Outcome' is a supporter of EDM usability issues and EDM domestication failure represented in Figure 8.4. Nodes 21-26 are supporters of the EM-PM responsibility conflict. Node 28 is a supporter of EDM domestication failure and practice culture. And finally Node 29 is a supporter of the EDM domestication failure and of the exemption culture.

Table 8.1 - 'Most Important' nodes identified during unaided analysis

Node ID and Description of Most Important Nodes
3. Domestication Responsibility : Domestication responsibility has fallen between the cracks
6. Practice Culture : Tendency to follow norms rather than procedures

7. EM_Outcome : Encouraging the use of EDM in a standardised manner
8. EM_Outcome : Pursuing an improvements strategy based on process standardisation
9. PM_Outcome : Each PM encourages the use of EDM according to their own practices
10. Eng_Outcome : EDM is perceived to be a source of frustration
21. EM_Responsibility : Implementing change / process improvements
22. PM_Responsibility : Delivering product on time, on budget in accordance to contractual obligations
23. PM_Responsibility : Meeting customer expectations
25. Eng_Responsibility : Design Systems / components
26. Eng_Responsibility : Meeting time, budget, and quality pressures
28. Eng_Outcome : Extent and use of EDM varies on project by project basis
29. PM_Outcome : Each PM implements and follows data management plans according to their own practices

Secondly, having identified the 'most important' nodes we generated eigenvector centrality scores and visually inspected the results to explore the relationships. We observed that the eight most highly ranked nodes (in Table 8.2) are members of the 'most important' group (Table 8.1). In other words there appeared to be a good correspondence between the 'most important' nodes as identified during unaided analysis (See Table 8.1) and those highly ranked according to eigenvector centrality (See Table 8.2).

Table 8.2 - Ranking of Node Importance using Eigenvector Centrality

#	Node ID and Description	EV Centrality
1	6. Practice Culture : Tendency to follow norms rather than procedures	1.00E+000
2	28. Eng_Outcome : Extent and use of EDM varies on project by project basis	9.54E-001
3	7. EM_Outcome : Encouraging the use of EDM in a standardised manner	8.51E-001
4	9. PM_Outcome : Each PM encourages the use of EDM according to their own practices	8.35E-001
5	3. Domestication Responsibility : Domestication responsibility has fallen between the cracks	6.19E-001
6	10. Eng_Outcome : EDM is perceived to be a source of frustration	4.38E-001
7	8. EM_Outcome : Pursuing an improvements strategy based on process standardisation	3.53E-001
7	29. PM_Outcome : Each PM implements and follows data management plans according to own practices	3.35E-001
9	30. EM_Objective : Pursue an improvements strategy based on process standardisation and Lean thinking	2.76E-001
10	16. PM_Objective : Meet customer expectations	1.07E-001
11	19. Eng_Objective : Meet programme managers schedule, cost and quality expectations	8.85E-002
12	15. PM_Objective : Meet contractual obligations	8.01E-002
13	31. Eng_Objective : Meet engineering managers process expectations	2.86E-002
14	23. PM_Responsibility : Meeting customer expectations	3.04E-003
15	18. Eng_Objective : Design high quality component or system	1.74E-003
16	22. PM_Responsibility : Delivering product on time, on budget in accordance to contractual obligations	1.29E-003
17	17. PM_Objective : Timely Internal reporting	2.75E-004

17	26. Eng_Responsibility : Meeting time, budget, and quality pressures	2.75E-004
17	12. EM_Objective : Improve quality	2.75E-004
17	13. EM_Objective : Improve safety	2.75E-004
17	14. EM_Objective : Improve efficiency	2.75E-004
17	21. EM_Responsibility : Implementing change / process improvements	2.75E-004
17	11. EM_Objective : Improve delivery time	2.75E-004
24	24. PM_Responsibility : Internal reporting	0.00E+000
24	25. Eng_Responsibility : Design Systems / components	0.00E+000
24	1. EDM : Document Management System	0.00E+000
24	20. EM_Responsibility : Improving delivery time, quality, safety, efficiency, repeatability	0.00E+000
24	4. Shared Folders : Document Management System	0.00E+000
24	27. Eng_Responsibility : Following process	0.00E+000
24	2. Matrix Structure : Engineers managed by Programme management & Engineering Manager	0.00E+000

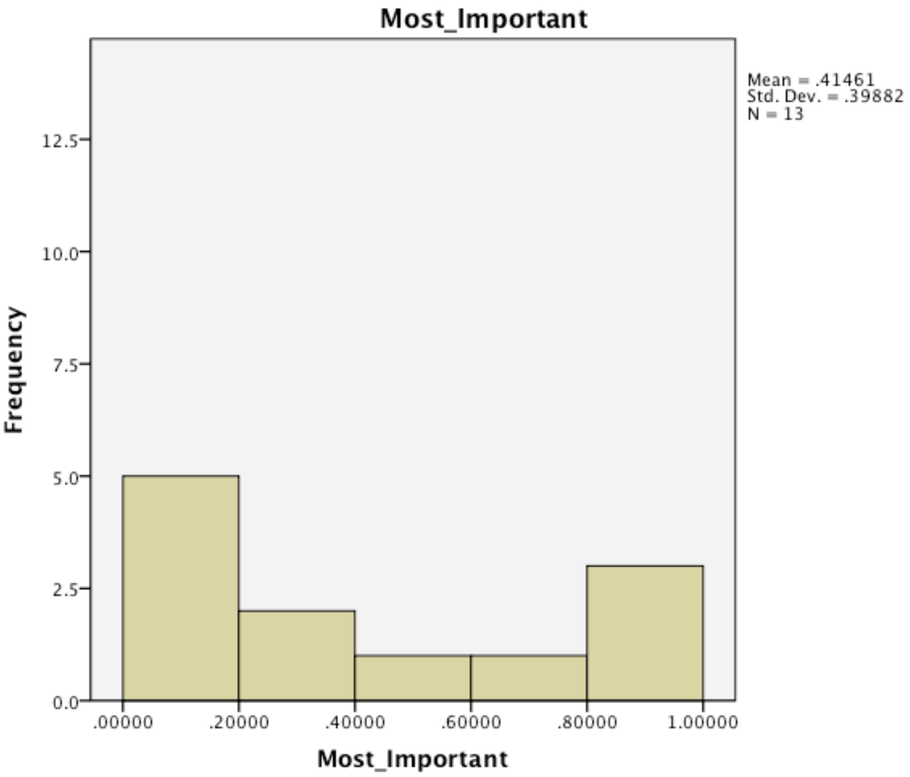
Statistical Test of Hypothesis 1

In order to test hypothesis 1 we performed an Independent-Samples Mann-Whitney U test using PASW. The results of the Mann-Whitney U test indicated to reject the null hypothesis with a p = 0.01. The distribution of eigenvector centrality scores was not the same across the population of ‘most important’ and ‘least important’ elements.

Table 8.3 - Descriptive Statistics of 'Most Important' and 'Least Important'

Statistics

		Most Important	Least Important
N	Valid	13	17
Mean		.4146062	.0343126
Median		.3530000	.0002750
Std. Deviation		.39882271	.07188513
Minimum		.00000	.00000
Maximum		1.00000	.27600



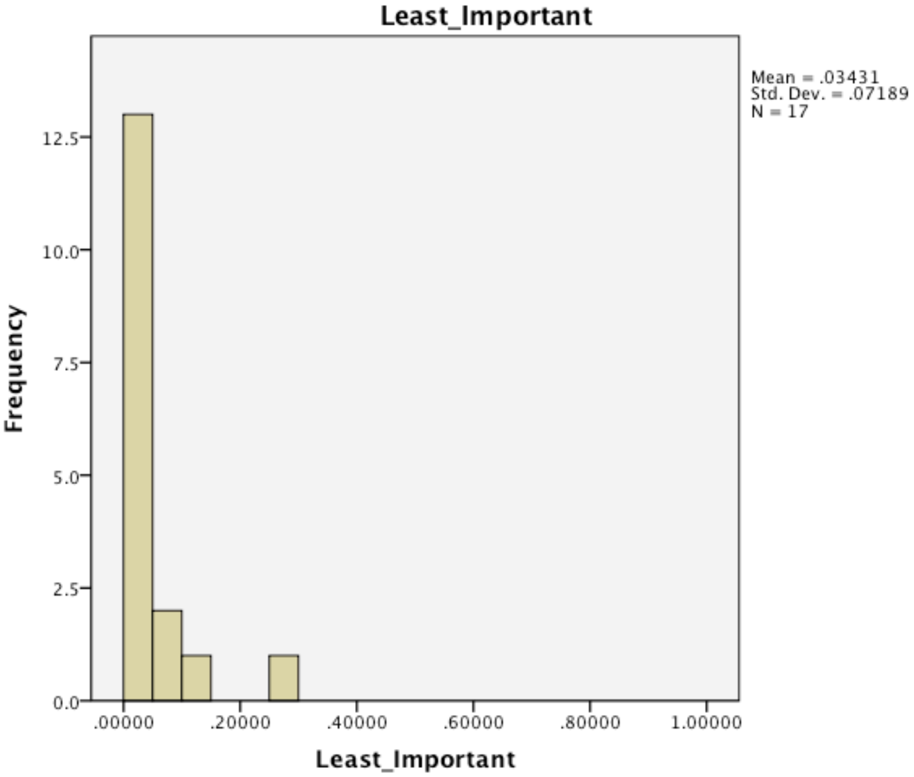


Figure 8.5 - Histograms of Most Important & Least Important

Via inspection of each population's descriptive statistics and histograms (See Table 8.3, Figure 8.5) one can corroborate the findings of the Mann-Whitney U test by observing that the standard deviation, mean, median, minimum and maximum values of the least important and most important populations are very different. The most important population had a mean of .41, a median of .35, a standard deviation of .40 and a minimum of 0 and a maximum of 1. In contrast the least important population had a small standard deviation and a mean and median close to zero with a minimum of 0 and maximum of 0.28.

The corroborated Mann-Whitney U test enables us to conclude that elements with a large eigenvector centrality are more likely to be a member of the 'most important' population than the least important population[14]. This confirms that ranking elements by eigenvector centrality is a reasonable indicator of the importance of an element in a problematic system. These findings thus indicate that computers may be used to aid the analysis of large problematic socio-technical systems by identifying elements that are most important to sustaining the current problematic system.

Statistical Test of Hypothesis 2

In order to test whether feedback degree is a good indicator of 'complexity' (hypothesis 2) we ranked (without the aid of network analysis) the 'complexity' of nodes in three subsections of the overall problematic system. The reason we used subsections is because the ranking of every node in the whole problematic system (with over 30 nodes and over 60 interactions) may have been unreliable due to human error. Therefore we opted for three chunks of the problematic system comprising 10 or fewer nodes. The subsections we selected were responsibilities and their interactions (see Figure 8.6 and Table 8.4), outcomes and their interactions (see Figure 8.7) and objectives and their interactions (see Figure 8.8). We selected these sub graphs as a convenience sample since they each had 10 or fewer nodes.

[14] One must of course be sensible when interpreting eigenvector centrality rankings as a higher score indicates an element is more *likely* to be a member of the 'most important' population but it does not mean it is *necessarily* a member of the 'most important' population.

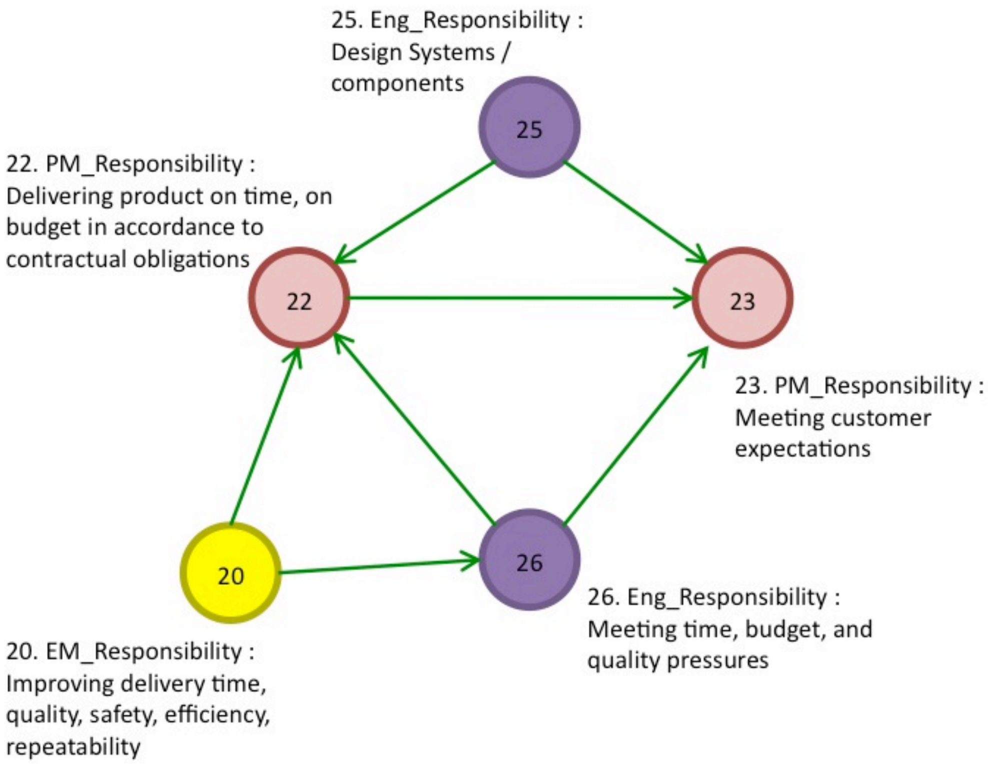

25. Eng_Responsibility : Design Systems / components
25
22. PM_Responsibility : Delivering product on time, on budget in accordance to contractual obligations
22
23
23. PM_Responsibility : Meeting customer expectations
20
26
26. Eng_Responsibility : Meeting time, budget, and quality pressures
20. EM_Responsibility : Improving delivery time, quality, safety, efficiency, repeatability

Figure 8.6 - Directed Graph of Responsibilities

Table 8.4 - Ranking of Complexity of Responsibility Nodes

Id	FD	Rank
22. PM_Responsibility : Delivering product on time, on budget in accordance to contractual obligations	0.4	1
23. PM_Responsibility : Meeting customer expectations	0.3	2
26. Eng_Responsibility : Meeting time, budget, and quality pressures	0.3	3
20. EM_Responsibility : Improving delivery time, quality, safety, efficiency, repeatability	0.2	4
25. Eng_Responsibility : Design Systems / components	0.2	4

Our rationale for ranking the complexity of responsibility elements (Figure 8.6) is as follows. One may observe that node 22 is involved in the most complex interactions in comparison to its peers. This may be observed by the fact that it is influenced by three nodes and influences another node. It helps the PM fulfil his responsibility to meet customer expectations (node 23) and is helped by the EM responsibility to improve delivery time, quality (and so on) (node 20) and the engineers responsibility to design components (node 25) and systems and to meet time, budget and quality pressures (node 26).

The second most highly ranked node is node 23 as it is influenced by three others nodes but does not influence any node. This may be observed by the fact that the fulfilment of the responsibility to meet customers expectations is helped by the engineers responsibility to meet time, budget and quality pressures (node 26), the PM's responsibility to deliver product on time on budget in accordance to contractual obligations (node 22), and engineers responsibility to design systems and components.

The third most highly ranked node is node 26 as it is influenced by one other node and influences two other nodes. The engineer's responsibility to meet time, budget and quality pressures is helped by EM responsibility to improve delivery time, quality, safety, efficiency, repeatability (node 20). Node 26 helps the PM's responsibility to delivery product on time and on budget in accordance with contractual obligations, and it also helps the PM's responsibility to meet customer expectations. The four most complex nodes are node 20 and 25. They both exhibit the least complex behaviour as they are not influenced by any other node but both influence two other nodes. A similar rationale was used for ranking the complexity of nodes in outcome (Figure 8.7) and objectives (Figure 8.8) sub graphs.

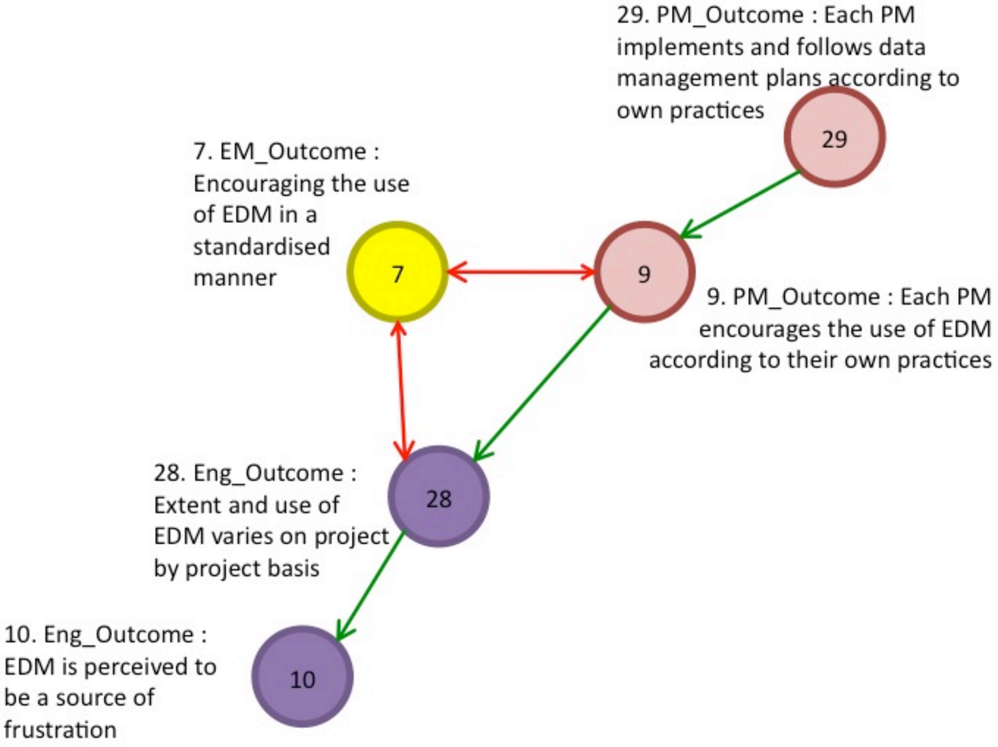

29. PM_Outcome : Each PM implements and follows data management plans according to own practices
29
7. EM_Outcome : Encouraging the use of EDM in a standardised manner
7
9
9. PM_Outcome : Each PM encourages the use of EDM according to their own practices
28. Eng_Outcome : Extent and use of EDM varies on project by project basis
28
10. Eng_Outcome : EDM is perceived to be a source of frustration
10

Figure 8.7 - Directed Graph of Outcomes

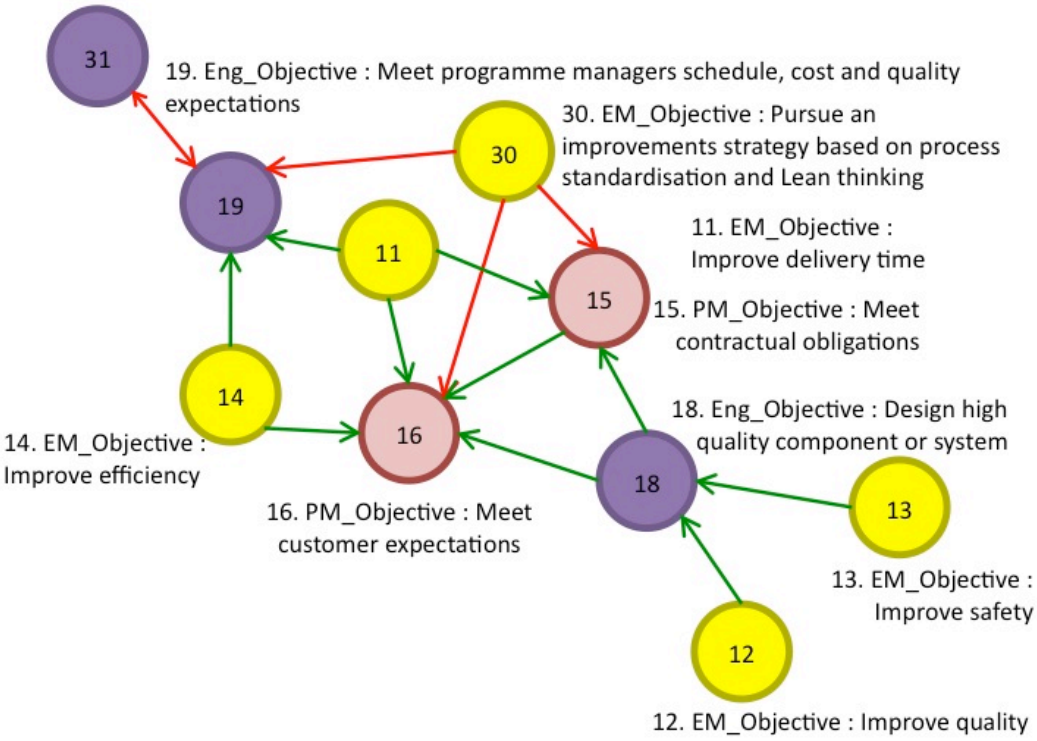

31
19. Eng_Objective : Meet programme managers schedule, cost and quality expectations
30
30. EM_Objective : Pursue an improvements strategy based on process standardisation and Lean thinking
19
11
11. EM_Objective : Improve delivery time
15
15. PM_Objective : Meet contractual obligations
14
16
18. Eng_Objective : Design high quality component or system
14. EM_Objective : Improve efficiency
18
13
16. PM_Objective : Meet customer expectations
13. EM_Objective : Improve safety
12
12. EM_Objective : Improve quality

Figure 8.8 - Directed Graph of Objectives

In order to test hypothesis 2, we performed Spearman's correlation test on the data from these three sub graphs using PASW. For all sub graphs tested we were able to detect statistically significant correlations between feedback degree (FD) and complexity as judged by an analyst. For the responsibility sub graph it may be observed that FD has a correlation of -0.973, which is statistically significant at the 0.01 level (Table 8.5). For the outcome sub graph it may be observed that FD has a correlation of -0.997, which is statistically significant at the 0.01 level (See Table 8.6). For the objectives sub graph it may be observed that FD has a correlation of -.975, which is statistically significant at the 0.01 level (See Table 8.7). This enables us to reject the null hypothesis. These findings confirm hypothesis 2 as the correlations are statistically significant account for a significant proportion of the variance e.g. r>0.83. These findings thus indicate that computers may be used to aid the analysis of large problematic socio-technical systems by identifying elements that display the most complex behaviour.

Table 8.5 Spearman's Rho for Responsibility Sub Graph

Correlations

			Degree	Loops	Feedback Degree
Spearman's rho	Rank	Correlation Coefficient	-.973**	.	**-.973****
		Sig. (2-tailed)	.005	.	.005
		N	5	5	5

**. Correlation is significant at the 0.01 level (2-tailed).

Table 8.6 - Spearman's Rho for Objectives Sub Graph

Correlations

			Degree	Loops	Feedback Degree
Spearman's rho	Rank	Correlation Coefficient	-.768**	-.701*	**-0.997****
		Sig. (2-tailed)	.010	.024	.010
		N	10	10	10

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Table 8.7 - Spearman's Rho for Outcome Sub Graph

Correlations

			Degree	Loops	Feedback Degree
Spearman's rho	Rank	Correlation Coefficient	-.866	-0.975**	**-0.975****
		Sig. (2-tailed)	.058	.005	.005
		N	5	5	5

**. Correlation is significant at the 0.01 level (2-tailed).

8.6 Conclusion

In this chapter we demonstrate that structural-intentional responsibility modelling may be a scalable technique for the socio-technical analysis of systems. This was shown by providing proof-of-concept that that tools for large-scale network analysis and visualisation may provide a promising avenue for a *scalable* form of structural-intentional responsibility analysis. Using a case study of an enterprise document management (EDM) system (from chapter 7) we demonstrated structural-intentional responsibility modelling's potential to scale to larger systems by showing that:

i) a problematic socio-technical system may be represented as a directed graph such that the elements in the system are represented as nodes, and interactions between nodes as edges;

ii) that *eigenvector centrality* may be used to rank the importance of elements in a system and that highly ranked elements correspond to those identified as important by a human analyst;

iii) the 'complexity' of the system, or a part of a system, may be characterised using a *feedback degree* score that provides an indication of the extent of feedback loops and the interconnectedness of elements.

These findings more generally indicate that computers may be used to aid the analysis of problematic large socio-technical systems within organisations by highlighting elements, or groups of interacting elements, that are important to the overall outcome of a system. This contribution is significant as it demonstrates the partial automation of a laborious process and opens an avenue for developing engineering techniques suited to troubleshooting systems where analysts cannot be expected to inspect each individual node due to their overwhelming number.

We hope this may be the starting point of a discussion within the socio-technical systems community as to how network analysis may aid the analysis and engineering of large socio-technical systems. There is a need for further work in this area. Of up most importance is the pursuit of further case studies to verify these early findings in other systems, contexts and domains. Whilst the analysis of problematic systems has been traditionally a qualitative process we believe as the scale of problematic systems increases the use of metrics to guide analysis becomes increasingly compelling.

Our conclusion is limited by the usual limitations of case study research. Case study research may not be generalisable and whilst every effort was taken to minimise investigator or participant bias, bias may be reflected in our findings.

9. Conclusion and Future Work

9.1 Conclusion

This thesis' overall conclusion is that the notion of 'responsibility' is a promising abstraction for representing and analysing systems that are composed of parts that are independently managed and maintained by agents spanning multiple organisational boundaries e.g. systems-of-systems, coalitions of systems, enterprise-scale systems.

It was argued in chapter 4 that the notion of *'responsibility'* provides a suitable abstraction for facilitating the enquiry, representation and understanding of systems by teasing out potentially intricate dependencies between technical and non-technical agents, whilst providing a suitable language for representing and discussing obligations, liabilities and norms which are important for understanding the threats and issues that may arise when systems span multiple organisational boundaries.

This argument was evidenced by means of qualitative case studies that illustrated how the responsibility abstraction can be used, or extended, to facilitate the analysis of real world problematic situations.

In chapter 5, responsibility modelling was used to enable the identification of socio-technical threats to the dependability of a coalition-of-systems. Coalitions-of-systems (CoS) are a class of system similar to SoS but they differ in that they interact to further overlapping self-interests rather than an overarching mission. Assessing the socio-technical dependability of CoS is an open research question of societal importance as existing socio-technical dependability analysis techniques typically do not assess threats associated with coalition partners reneging on responsibilities or leaving a coalition. We used a cloud computing based case study to demonstrate that a responsibility modelling based threat identification approach enables the identification of these threats. This research provided first evidence that inspecting the distribution of liabilities among coalition partners may indicate the fragility of overlapping self-interests.

In chapter 6, responsibility modelling was extended to identify agent-related threats. Current approaches to threat identification view stakeholders from a mechanistic means-end perspective where human agents are assumed to be passive and compliant. In this chapter, we used conflict theory to develop a framework to assess threats to cooperative agent behaviour. We made the case that this class of risk is an important class that is missing from threat identification approaches. We extended the cloud computing based case study from chapter 5 to demonstrate that conflict based threat identification enables the identification of threats to cooperative behaviour in SoS type situations.

In chapter 7, a responsibility-based troubleshooting approach was developed building upon the work in chapter 6. In this chapter it was postulated that ethnographic and social analyses have not been widely assimilated into industry practice because they did not fit practitioners' practices. In response to this, we developed a lightweight qualitative responsibility-based technique to provide insights to ameliorate problematic enterprise-scale system deployments. Unlike typical ethnographies and social analyses of *work activity* that inform systems *analysis* and *design*; we argued that analysis of *intentional* and *structural factors* to inform *system deployment* and *integration* could have a shorter time duration and yet can provide actionable insights. We evaluated our approach using a case study of a problematic enterprise document management system within a multinational systems engineering organisation. Our findings are of academic and practical significance as our approach demonstrated that structural-intentional analysis scales to enable the timely analysis of enterprise system deployments.

In chapter 8, it was argued that techniques for network analysis may enable the analysis of problematic enterprise-scale socio-technical systems comprising large numbers of nodes. By means of re-analysing the case study in chapter 7, we demonstrated a proof-of-concept responsibility-based technique for SoS scale network analysis and visualisation that may provide a promising avenue for identifying problematic elements and interactions amongst an overwhelming number of socio-technical elements. We demonstrated the potential of this approach by showing that: i) a problematic situation may be represented as a directed graph such that the elements in the situation are represented as nodes, and interactions between nodes as edges; ii) that eigenvector centrality may be used to rank the importance of elements in a situation and that highly ranked elements match those identified as important by a human analyst; iii) the 'complexity' of a situation, or a part of a situation, may be characterised using a feedback degree score which provides an indication of the extent elements are highly interconnected and involved in feedback loops. These findings indicated that computers could be used to aid the analysis of SoS situations by highlighting elements, or groups of interacting elements, that are important to the overall outcome of a problematic situation.

9.2 Threats to the Validity

This thesis adopted a mainly qualitative case study approach so the validity of these findings is largely dependent upon the plausibility of the phenomena of study being representative of SoS. Another limitation of this research, like all qualitative research, is that the findings rely upon the expertise of the researcher not to inject bias into the results. Reasonable measures were taken to control for bias, for example by checking interpretations with interview subjects and by presenting findings to domain experts for sense checking. However one may not rule out the possibility of bias influencing the findings. Further research by independent researchers is therefore required to validate this thesis' findings.

9.3 Future Work

The socio-technical analysis of SoS is still in a nascent form. Significant amounts of further work are required to transform this area into a mature field.

An area of primary importance is the *sensitisation* of practicing engineers to the socio-technical problems associated with systems that span organisational boundaries. Providing practitioners with the tools to 'see' these problems in their own systems is paramount because at present it seems that experienced practitioners have an intuition that something isn't quite right but they are unable to 'see' the problem until a researcher, like myself, casts their eye on the situation. The forms of responsibility modelling described in this thesis may be good avenues to approach sensitisation. They appear to be simple, practical and scalable. Further case studies testing these claims of practicality and scalability would be valuable to their development and extension. The exploration of the benefits and drawbacks of combining responsibility models with statistical models such as Bayesian belief networks, instead of causal diagrams, may also be of interest.

Another potentially valuable area of research related to *sensitisation* is the development of a taxonomy of pathogenic socio-technical system design patterns. A taxonomy could help reveal patterns, or common traits among systems, thus providing the equivalent of a 'periodic table' for socio-technical systems. Compiling an initial taxonomy from existing case studies and pursuing new cases would contribute to this area.

Another area of importance is the *amelioration* of problematic socio-technical systems. Once practitioners 'see' specific problems in their systems, 'solutions' are required. The development of a body of knowledge to help practitioners identify 'solutions' and understand their trade-offs would be valuable. Compiling a survey of existing work and pursuing action research would contribute to advancing this area.

To further research in socio-technical systems as a whole, an ambitious researcher may consider using new research methodologies. Whilst qualitative case study research is highly valuable for producing exploratory work, it limitations with respect to repeatability and validity may hinder advancements in the long term. Ambitious researchers may like to consider using case studies in tandem with a simulation environment, so as to be able to generate and manipulate observed phenomena 'in-silico'[15]. The process of generating a phenomenon 'in-silico' provides a rigorous means of testing theories by verifying whether they are indeed able to reproduce a phenomenon in a controlled environment. The use of simulation for this purpose is gaining traction in the social sciences and may be useful to the study of socio-technical systems (Epstein, 2007).

[15] The term 'in-silico' refers to the use of computers to reproduce (simulate) observed phenomena by means of representing the mechanisms for its generation.

Thesis References

Ackermann, F., Eden, C., Williams, T., & Howick, S. (2007). Systemic Risk Assessment: A Case Study. The Journal of the Operational Research Society, 58(1), 39-51.

Adams, S., & Berg, M. (2004). The nature of the Net: constructing reliability of health information on the Web. Information Technology & People, 17(2), 150-170.

Ajzen, I. (1991). The theory of planned behavior. Organizational Behavior and Human Decision Processes, 50(2), 179-211.

Allen, T. (2011). Introduction to Discrete Event Simulation and Agent-based Modeling: Voting Systems, Health Care, Military, and Manufacturing. London: Springer.

Altiok, T., & Melamed, B. (2007). Simulation Modeling and Analysis with ARENA. London: Elsevier.

Anderson, S., Hardstone, G., Procter, R., & Williams, R. (2005). Down in the (Data)base(ment): Supporting Configuration in Organisational Information Systems. In M. S. Ackerman, C. A. Halverson, T. Erickson & W. A. Kellogg (Eds.), Resources, Co-Evolution and Artifacts. London: Springer-Verlag.

Armitage, C. J., & Conner, M. (2001). Efficacy of the Theory of Planned Behaviour: A meta-analytic review. British Journal of Social Psychology, 40(4), 471-499.

Asnar, Y. (2009). Requirements Analysis and Risk Assessment for Critical Information Systems. Universita Degli Studi Di Trento.

Asnar, Y., & Giorgini, P. (2006). Ensuring Dependability in Socio-Technical System by Risk Analysis. Paper presented at the 6th European Dependable Computing Conference.

Asnar, Y., & Giorgini, P. (2007). From Trust to Dependability through Risk Analysis. Paper presented at the The Second International Conference on Availability, Reliability and Security.

Asnar, Y., Moretti, R., Sebastianis, M., & Zannone, N. (2008). Risk as Dependability Metrics for the Evaluation of Business Solutions: A Model-driven Approach. Paper presented at the Proceedings of the 2008 Third International Conference on Availability, Reliability and Security.

Assael, H., & Keon, J. (1982). Nonsampling vs. Sampling Errors in Survey Research. The Journal of Marketing, 46(2), 114-123.

Avison, D., Andrews, J. K., & Shah, H. U. (1992). Towards an SSM toolkit: rich picture diagramming. European Journal of Information Systems, 1(6).

Avison, D., & Fitzgerald, G. (2006). Information Systems Development: Methodologies, Techniques and Tools: McGraw-Hill Higher Education.

Avison, D., Wood-Harper, T., Vidgen, R., & Wood, J. R. G. (1998). A further exploration into information systems development: the evolution of Multiview2. Information Technology & People, 11(2), 124-139.

Avizienis, A., Laprie, J. C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. Dependable and Secure Computing, IEEE Transactions on, 1(1), 11-33.

Baskerville, R., & Pries-Heje, J. (1999). Grounded action research: a method for understanding IT in practice. Accounting, Management and Information Technologies, 9(1), 1-23.

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: An Open Source Software for Exploring and Manipulating Networks. Paper presented at the 3rd Int'l AAAI Conference on Weblogs and Social Media.

Bennett, S., Skelton, J., & Lunn, K. (2005). UML (Second Edition ed.). London: McGraw-Hill Education.

Beynon-Davies, P. (1995). Information systems 'failure': the case of the London Ambulance Service's Computer Aided Despatch project. European Journal of Information Systems, 4(3), 171-184.

Beynon-Davies, P. (1999). Human error and information systems failure: the case of the London ambulance service computer-aided despatch system project. Interacting with Computers, 11(6), 699-720.

Blyth, A. J. C., Chudge, J., Dobson, J. E., & Strens, M. R. (1993). ORDIT: a new methodology to assist in the process of eliciting and modelling organizational requirements. Paper presented at the Proceedings of the conference on Organizational computing systems.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., & Hwang, D. U. (2006). Complex networks: Structure and dynamics. Physics Reports, 424(4-5), 175-308.

Boland, R. J., Newman, M., & Pentland, B. T. (2010). Hermeneutical exegesis in information systems design and use. Information and Organization, 20(1), 1-20.

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. Journal of Mathematical Sociology(2), 113-120.

Borgatti, S. P. (2005). Centrality and network flow. Social Networks, 27(1), 55-71.

Braber, F. d., Dimitrakos, T., Gran, B. A., Lund, M. S., Stølen, K., & Aagedal, J. Ø. (2003). The CORAS methodology: model-based risk assessment using UML and UP. UML and the unified process (pp. 332-357): IGI Publishing.

Brown, R. (2000). Group Processes (2 ed.). Oxford: Blackwell.

Butler, T. (2003). An institutional perspective on developing and implementing intranet- and internet-based information systems. Information Systems Journal, 13(3), 209-231.

Butler, T., & Fitzgerald, B. (1997). A case study of user participation in the information systems development process. Paper presented at the Proceedings of the eighteenth international conference on Information systems.

Bygstad, B., Nielsen, P. A., & Munkvold, B. E. (2010). Four integration patterns: a socio-technical approach to integration in IS development projects. Information Systems Journal, 20(1), 53-80.

Checkland, P. (1999). Systems Thinking, Systems Practice Includes a 30 year Retrospective. Chicester: John Wiley & Sons.

Checkland, P., & Holwell, S. (1998). Information, Systems and Information Systems: Making Sense of the Field: John Wiley & Sons.

Cohen, C. F., Birkin, S. J., Garfield, M. J., & Webb, H. W. (2004). Managing conflict in software testing. Communications of the ACM, 47(1), 76-81.

Cole, M., & Avison, D. (2007). The potential of hermeneutics in information systems research. European Journal Information Systems, 16(6).

Committee, H. o. C. P. A. (2009). The National Programme for IT in the NHS: Progress since 2006. London: House of Commons.

Corbin, J., & Strauss, A. (2008). Basics of qualitative research : techniques and procedures for developing grounded theory. London: Sage Publications.

Corea, S. (2006). Mounting effective IT based customer service operations under emergent conditions: Deconstructing myth as a basis of understanding. Information and Organization, 16(2), 109-142.

Creswell, J. (2009). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. London: Sage.

Currie, W. L., & Guah, M. W. (2007). Conflicting institutional logics: a national programme for IT in the organisational field of healthcare. J Inf technol, 22(3), 235-247.

De Dreu, C. K. W., & McCusker, C. (1997). Gain-loss frames and cooperation in two-person social dilemmas: A transformational analysis. Journal of Personality and Social Psychology, 72(5), 1093-1106.

de Vreede, G.-J., & van Eijck, D. T. T. (1998). Modeling and Simulating Organizational Coordination. Simulation & Gaming, 29(1), 60-87.

Delone, W. H., & McLean, E. R. (2003). The DeLone and McLean Model of Information Systems Success: A Ten-Year Update. Journal of Management Information Systems, 19(4), 9-30.

Deutsch, M. (1949). An experimental study of the effects of cooperation and competition upon group process. . Human Relations, , 2, 199-231.

Deutsch, M. (1973). The Resolution of Conflict: Constructive and Destructive Processes American Behavioral Scientist, 17(2).

Deutsch, M. (1990). Sixty Years of Conflict. International Journal of Conflict Management, 1(3), 237-263.

Deutsch, M., & Coleman, P. T. (2006). The Handbook of Conflict Resolution: Theory and Practice (2nd Edition edition ed.): Jossey Bass.

Dickerson, C., & Mavris, D. (2009). Architecture and Principles of Systems Engineering CRC Press.

Dobson, J. (2007). Understanding Failure: The London Ambulance Service Disaster. In G. Dewsbury & J. Dobson (Eds.), Responsibility and Dependable Systems. London: Springer Verlag.

Dobson, J., & Martin, M. (2007). Models for Understanding Responsibilities. In G. Dewsbury & J. Dobson (Eds.), Responsibility and Dependable Systems. London: Springer-Verlag.

Doherty, N., & King, M. (2005). From technical to socio-technical change: tackling the human and organizational aspects of systems development projects. European Journal of Information Systems, 14(1-5).

Eden, C. (1988). Cognitive mapping. European Journal of Operational Research, 36(1), 1-13.

Eden, C. (2004). Analyzing cognitive maps to help structure issues or problems. European Journal of Operational Research, 159(3).

Eden, C., & Ackermann, F. (2004). Cognitive mapping expert views for policy analysis in the public sector. European Journal of Operational Research, 152(3), 615-630.

Eden, C., Ackermann, F., & Cropper, S. (1992). The Analysis of Cause Maps. Journal of Management Studies, 29(3), 309-324.

Ellingsen, G., & Monteiro, E. (2003). A Patchwork Planet Integration and Cooperation in Hospitals. Computer Supported Cooperative Work, 12(1), 71-95.

Engestrom, Y. (1996). The tensions of judging: handling cases of driving under the influence of alcohol in Finland and California. In Y. Engestrom & D. Middleton (Eds.), Cognition and Communication at Work. Cambridge: Cambridge University Press.

Engestrom, Y. (1999). Activity theory and individual and social transformation. In Y. Engestrom, R. Miettinen & R.-L. Punamaki (Eds.), Perspectives on activity theory. Cambridge: Cambridge University Press.

Engestrom, Y. (2000). Activity theory as a framework for analyzing and redesigning work. Ergonomics, 43(7), 960.

Engestrom, Y. (2001). Expansive Learning at Work: toward an activity theoretical reconceptualization. Journal of Education and Work, 14, 133-156.

Epstein, J. M. (2007). Generative Social Science: Studies in Agent-Based Computational Modeling. New Jersey: Princeton University Press.

Faraj, S., Kwon, D., & Watts, S. (2004). Contested artifact: technology sensemaking, actor networks, and the shaping of the Web browser. Information Technology & People, 17(2), 186 - 209.

Finkelstein, A., & Dowell, J. (1996). A comedy of errors: the London Ambulance Service case study. Paper presented at the Proceedings of the 8th International Workshop on Software Specification and Design.

Fitzgerald, G., & Russo, N. L. (2005). The turnaround of the London ambulance service computer-aided despatch system (LASCAD). European Journal of Information Systems, 14(3), 244-257.

Forrester, J. (1958). Industrial Dynamics: A Major Breakthrough for Decision Makers. Harvard Business Review, 36(4), 37-66.

Forrester, J. (1971). Counterintuitive behavior of social systems. Theory and Decision, 2(2), 109-140.

Gane, C., & Satson, T. (1979). Structured Systems Analysis: Tools and Techniques. New Jersey: Prentice Hall.

Gans, G., Jarke, M., Kethers, S., Lakemeyer, G., & Schmitz, D. (2011). Requirements Engineering for Trust-Based Interorganisational Networks. In E. Yu, P. Giorgini, N. Maiden & J. Mylopoulos (Eds.), Social Modeling for Requirements Engineering. Cambridge: MIT Press.

Gansner, E. R., & North, S. C. (2000). An open graph visualization system and its applications to software engineering. Softw. Pract. Exper., 30(11), 1203-1233.

Garlan, D., Allen, R., & Ockerbloom, J. (2009). Architectural Mismatch: Why Reuse Is Still So Hard. IEEE Software, 26(4), 66-69.

Gorod, A., Sauser, B., & Boardman, J. (2008). System-of-Systems Engineering Management: A Review of Modern History and a Path Forward. Systems Journal, IEEE, 2(4), 484-499.

Greenwood, D., & Sommerville, I. (2011a). Expectations and Reality: Why an enterprise software system didn't work as planned. Paper presented at the 20th Conference on Information Systems Development.

Greenwood, D., & Sommerville, I. (2011b). Responsibility Modeling for Identifying Sociotechnical Threats to the Dependability of Coalitions of Systems. Paper presented at the 6th IEEE International Conference on Systems of Systems Engineering (SoSE).

Greenwood, D., & Sommerville, I. (2011c). Using Complex Network Analysis And Visualisation To Analyse Problematic Enterprise Scale Information Systems? Paper presented at the 55th Meeting of the International Society for the Systems Sciences.

Greer, L. L., & Jehn, K. A. (2007). Chapter 2 The Pivotal Role of Negative Affect in Understanding the Effects of Process Conflict on Group Performance (Vol. 10): Emerald Group Publishing Limited.

Guizzardi, R., Perini, A., & Dignum, V. (2011). Socially Grounded Analysis of Knowledge Management Systems and Processes. In E. Yu, G. Paolo, N. Maiden & J. Mylopoulos (Eds.), Social Modeling for Requirements Engineering. Cambridge: MIT Press.

Hair, J., Black, W., Babin, B., & Anderson, R. (2008). Multivariate Data Analysis: A Global Perspective (7th Edition ed.). New Jersey: Pearson Education.

Halloran, J. (2000). The Activity Space: Analyzing Intentionality in Open Cooperative Work. . University of Sussex, Sussex UK.

Halloran, J. (2001). Taking the 'No' out of Lotus Notes: activity theory, groupware, and student groupwork. Paper presented at the CSCL '02, International Conference on Computer-supported Collaborative Learning.

Hartmann, A., & Bresnen, M. (2011). The emergence of partnering in construction practice: an activity theory perspective. Engineering Project Organization Journal, 1(1), 41 - 52.

Health, D. o. (2011). Dismantling the NHS National Programme for IT. Retrieved 29/01/2012, 2012, from http://mediacentre.dh.gov.uk/2011/09/22/dismantling-the-nhs-national-programme-for-it

Hine, C. (2007). Connective Ethnography for the Exploration of e-Science. Journal of Computer-Mediated Communication, 12(2), 618-634.

Hirschheim, R., & Newman, M. (2002). Symbolism and Information Systems Development: Myth, Metaphor and Magic. In M. Myers & D. Avison (Eds.), Qualitative Research in Information Systems. London: Sage.

Hjortso, C. N., Christensen, S. M., & Tarp, P. (2005). Rapid stakeholder and conflict assessment for natural resource management using cognitive mapping: The case of Damdoi Forest Enterprise, Vietnam. Agriculture and Human Values, 22(2), 149-167.

Hollnagel, E. (2004). Barriers And Accident Prevention. Aldershot: Ashgate Publishing Company.

Hollnagel, E. (2008). Critical Information Infrastructures: Should Models Represent Structures or Functions? Paper presented at the Proceedings of the 27th international conference on Computer Safety, Reliability, and Security.

Hollnagel, E., & Goteman, O. (2004). The Functional Resonance Accident Model. Paper presented at the Proceedings of Cognitive System Engineering in Process Plant 2004.

Hollnagel, E., Pruchnicki, S., Woltjer, R., & Etcher, S. (2008). Analysis of Comair flight 5191 with the functional resonance accident model. Paper presented at the 8th International Symposium of the Australian Aviation Psychology Association.

Horkoff, J., & Yu, E. (2009). Evaluating Goal Achievement in Enterprise Modeling: An Interactive Procedure and Experiences The Practice of Enterprise Modeling. In A. Persson & J. Stirna (Eds.), (Vol. 39, pp. 145-160): Springer Berlin Heidelberg.

Horkoff, J., & Yu, E. (2010). Finding Solutions in Goal Models: An Interactive Backward Reasoning Approach Conceptual Modeling. In J. Parsons, M. Saeki, P. Shoval, C. Woo & Y. Wand (Eds.), (Vol. 6412, pp. 59-75): Springer Berlin / Heidelberg.

Hougham, M. (1996). London Ambulance Service computer-aided despatch system. International Journal of Project Management, 14(2), 103-110.

Howard, P. N. (2002). Network Ethnography and the Hypermedia Organization: New Media, New Organizations, New Methods. New Media & Society, 4(4), 550-574.

Hubbard, D. (2010). How To Measure Anything: Finding the Value of Intangibles in Business. Hoboken, New Jersey: John Wiley & Sons.

Hughes, J., & Jones, S. (2003). Reflections on the Use of Grounded Theory in Interpretive Information Systems Research. Paper presented at the European Conference on Information Systems.

Hughes, J., King, V., Rodden, T., & Andersen, H. (1995). The role of ethnography in interactive systems design. Interactions, 2(2), 56-65.

Hutchins, E. (1995). How a cockpit remembers its speeds. Cognitive Science, 19(3), 265-288.

Imran, S., Foping, F., Feehan, J., & Dokas, I. M. (2010). Domain Specific Modeling Language for Early Warning System: Using IDEF0 for Domain Analysis. International Journal of Computer Science Issues, 7(5), 10-17.

Isssroff, K., & Scanlon, E. (2002). Using technology in Higher Education: an Activity Theory perspective. Journal of Computer Assisted Learning, 18(1), 77-83.

Jamshidi, M. (2008). Introduction to system of systems. In M. Jamshidi (Ed.), Systems of Systems Engineering: Principles and Applications (pp. 1-36). Boca Raton, Florida: CRC Press.

Janssen, O., Vliert, E. V. D., & Veenstra, C. (1999). How Task and Person Conflict Shape the Role of Positive Interdependence in Management Teams. Journal of Management, 25.

Jehn, K. A. (1995). A Multimethod Examination of the Benefits and Detriments of Intragroup Conflict. Administrative Science Quarterly, 40(2), 256-282.

Jehn, K. A. (1997). A Qualitative Analysis of Conflict Types and Dimensions in Organizational Groups. Administrative Science Quarterly, 42(3), 530-557.

Jehn, K. A., & Mannix, E. A. (2001). The Dynamic Nature of Conflict: A Longitudinal Study of Intragroup Conflict and Group Performance. The Academy of Management Journal, 44(2), 238-251.

Joshi, K., & Rai, A. (2000). Impact of the quality of information products on information system users' job satisfaction: an empirical investigation. Information Systems Journal, 10(4), 323-345.

Kaplan, B., & Maxwell, J. (2005). Qualitative Research Methods for Evaluating Computer Information Systems Evaluating the Organizational Impact of Healthcare Information Systems. In J. Anderson & C. Aydin (Eds.), (pp. 30-55): Springer New York.

Kaptelinin, V., & Nardi, B. (2006). Acting with Technology: Activity Theory and Interaction Design. Cambridge, Massachusetts: The MIT Press.

Kerr, N. L., & Kaufman-Gilliland, C. M. (1994). Communication, commitment, and cooperation in social dilemma. Journal of Personality and Social Psychology, 66(3), 513-529.

King, P., & Pooley, R. (2000). Derivation of Petri Net Performance Models from UML Specifications of Communications Software Computer Performance Evaluation.Modelling Techniques and Tools. In B. Haverkort, H. Bohnenkamp & C. Smith (Eds.), (Vol. 1786, pp. 262-276): Springer Berlin / Heidelberg.

Kock, N., & McQueen, R. J. (1998). Groupware support as a moderator of interdepartmental knowledge communication in process improvement groups: an action research study. Information Systems Journal, 8(3), 183-198.

Lash, S. (2001). Technological Forms of Life. Theory, Culture & Society, 18(1), 105-120.

Latour, B. (2005). Reassembling the Social: An Introduction to Actor-Network-Theory. Oxford: Oxford University Press.

Lawrence, K. A. (2006). Walking the Tightrope: The Balancing Acts of a Large e-Research Project. Computer Supported Cooperative Work, 15(4), 385-411.

Lazar, J., Feng, J. H., & Hochheiser, H. (2010). Research Methods In Human-Computer Interaction (First ed.). Chichester: John Wiley & Sons.

Lee, A. S. (1994). Electronic Mail as a Medium for Rich Communication: An Empirical Investigation Using Hermeneutic Interpretation. MIS Quarterly, 18(2), 143-157.

Lee, A. S. (2004). Thinking about Social Theory and Philosophy for Information Systems. In J. Mingers & L. Willcocks (Eds.), Social Theory and Philosophy for Information Systems (pp. 1-26). Chichester: John Wiley & Sons.

Leveson, N. (2004). A new accident model for engineering safer systems. Safety Science, 42(4), 237-270.

Leveson, N., Daouk, M., Dulac, N., & Marais, K. (2003). Applying STAMP in Accident Analysis. Paper presented at the Workshop on the Investigation and Reporting of Accidents.

Leveson, N., Dulac, N., Zipkin, D., Cutcher-Gershenfeld, J., Carrol, J., & Barrett, B. (2006). Engineering Resilience into Safety-Critical Systems. In E. Hollnagel, D. Woods & N. Leveson (Eds.), Resilience Engineering: Concepts and Precepts. Aldershot: Ashgate.

Li, G., & Yao, S. (2009, 19-21 May 2009). Research on Mapping Algorithm of UML Sequence Diagrams to Object Petri Nets. Paper presented at the Intelligent Systems, 2009. GCIS '09. WRI Global Congress on.

Lock, R., Storer, T., & Sommerville, I. (2009). Responsibility Modelling for Risk Analysis. Paper presented at the ESREL 2009.

Mähring, M., Holmström, J., Keil, M., & Montealegre, R. (2004). Trojan actor-networks and swift translation: Bringing actor-network theory to IT project escalation studies. Information Technology & People, Vol. 17 (2), 210 - 238.

Maiden, N., & Jones, S. (2004). Dependability in RESCUE: A Concurrent Engineering Approach to the Specification of Requirements for Air Traffic. Paper presented at the The Dependability, Systems and Networks workshop on interdisciplinary approaches.

Maiden, N., Kamdar, N., & Bush, D. (2006). Analyzing I* System Models for Dependability Properties: The Uberlingen Accident. Paper presented at the The 12th International Workshop on Requirements Engineering: Foundation For Software Quality.

Maier, M. W. (1998). Architecting Principles for System of Systems. Systems Engineering, 1(4), 267-284.

Marres, N. (2004). Tracing the trajectories of issues, and their democratic deficits, on the Web: The case of the Development Gateway and its doubles". Information Technology & People, 17(2), 124 - 149.

Matavire, R., & Brown, I. (2008). Investigating the use of "Grounded Theory" in information systems research. Paper presented at the Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology.

Mayer, N., Dubois, E., & Rifaut, A. (2007). Requirements Engineering for Improving Business/IT Alignment in Security Risk Management Methods. In R. J. Gonçalves, J. P. Müller, K. Mertins & M. Zelm (Eds.), Enterprise Interoperability II (pp. 15-26): Springer London.

McCarter, B. G., & White, B. E. (2008). Emergence of SoS, sociocognitive aspects. In M. Jamshidi (Ed.), Systems of Systems Engineering: Principles and Applications (pp. 71-102). Boca Raton, Florida: CRC Press.

Melão, N., & Pidd, M. (2000). A conceptual framework for understanding business processes and business process modelling. Information Systems Journal, 10(2), 105-129.

Millen, D. R. (2000). Rapid ethnography: time deepening strategies for HCI field research. Paper presented at the Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques.

Miller, J., & Page, S. (2007). Complex Adaptive Systems: An Introduction to Computational Models of Social Life. New Jersey: Princeton University Press.

Mingers, J., & Taylor, S. (1992). The Use of Soft Systems Methodology in Practice. The Journal of the Operational Research Society, 43(4), 321-332.

Moeller, G., & Zhang, X. (2008). Understanding antecedents of interpersonal conflict in information systems development: A critical analysis. Paper presented at the Annual Meeting of the Decision Sciences Institute, Baltimore Maryland.

Montibeller, G., & Belton, V. (2006). Causal Maps and the Evaluation of Decision Options: A Review. The Journal of the Operational Research Society, 57(7), 779-791.

Mumford, E. (1995). Effective Systems Design and Requirements Analysis: The Ethics approach: Palgrave.

Myers, M. (1997). Qualitative Research in Information Systems. MIS Quarterly, 21(2), 241-242.

Myers, M. (2004). Hermeneutics in Information Systems Research. In J. Mingers & L. Willcocks (Eds.), Social Theory and Philosophy for Information Systems (pp. p103-128). Chicester: John Wiley & Sons.

Myers, M., & Avison, D. (2002). An Introduction to Qualitative Research in Information Systems. In M. Myers & D. Avison (Eds.), Qualitative Research in Information Systems (pp. p3-12). London: Sage.

Myers, M., & Young, L. (1997). Hidden Agendas, power and managerial assumptions in information systems development: An ethnographic study. Information Technology & People, 10(3).

NAO. (2011). The National Programme for IT in the NHS: an update on the delivery of detailed care records systems: National Audit Office.

Newman, M. E. J. (2003). The Structure and Function of Complex Networks. SIAM Review, 45(2), 167-256.

NIST, N. I. o. S. a. T. (1993). Draft Federal Information Processing Standard Publication 183, Integration Definition for Function Modeling (IDEF0). Springfield, VA National Technical Information Service.

Northrop, L., Feiler, P., Gabriel, R. P., Goodenough, J., Linger, R., Longstaff, T., et al. (2006). Ultra-Large-Scale Systems: The Software Challenge of the Future: Carnegie Mellon University Software Engineering Institute.

Office, C. (2011). MAJOR PROJECTS AUTHORITY PROGRAMME ASSESSMENT REVIEW of the National Programme for IT. London: Cabinet Office.

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), 245-251.

Orlikowski, W. J. (1993). CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development. MIS Quarterly, 17(3), 309-340.

Orlikowski, W. J., Yates, J., Okamura, K., & Fujimoto, M. (1995). Shaping Electronic Communication: The Metastructuring of Technology in the Context of Use. Organization Science, 6(4), 423-444.

Page, D., Williams, P., & Boyd, D. (1993). Report of the Inquiry Into The London Ambulance Service. London: The Communications Directorate, South West Thames Regional Health Authority.

Pastor, O., Estrada, H., & Martinez, A. (2011). Strengths and Weaknesses of the i* Framework: An Empirical Evaluation. In E. Yu, P. Giorgini, N. Maiden & J. Mylopoulos (Eds.), Social Modelling for Requirements Engineering. Cambridge, Massachusetts: MIT Press.

Peters, L., & Peters, J. (1997, 20-25 Apr 1997). Using IDEF0 for dynamic process analysis. Paper presented at the Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on.

Plaia, A., & Carrie, A. (1995). Application and assessment of IDEF3-process flow description capture method. International Journal of Operations & Production Management, 15(1), 63 - 73.

Pollock, N., & Williams, R. (2010). e-Infrastructures: How Do We Know and Understand Them? Strategic Ethnography and the Biography of Artefacts. Computer Supported Cooperative Work (CSCW), 19(6), 521-556.

Pons, D., & Raine, J. (2005). Design mechanisms and constraints. Research in Engineering Design, 16(1), 73-85.

Pooley, R., & King, P. (1999). The Unified Modelling Language and performance engineering. Software, IEE Proceedings -, 146(1), 2-10.

Pyster, A., D. Olwell, J. Anthony, S. Enck, N. Hutchison, and A. Squires. (2011). A Guide to the Systems Engineering Body of Knowledge (SEBoK). Hoboken, NJ: Stevens Institute of Technology.

RAE. (2004). The Challenges of Complex IT Projects: Royal Academy of Engineering.

Ramduny-Ellis, D., & Dix, A. (2007). Modelling in Practice. In G. Dewsbury & J. Dobson (Eds.), Responsibility and Dependable Systems. London: Springer-Verlag.

Rizzo, J. R., House, R. J., & Lirtzman, S. I. (1970). Role Conflict and Ambiguity in Complex Organizations. Administrative Science Quarterly, 15(2), 150-163.

Robey, D., Farrow, D. L., & Franz, C. R. (1989). Group Process and Conflict in System Development. Management Science, 35(10), 1172-1191.

Robey, D., Welke, R., & Turk, D. (2001). Traditional, iterative, and component-based development: A social analysis of software development paradigms. Information Technology and Management, 2, 53-70.

Romero, F., Company, P., Agost, M.-J., & Vila, C. (2008). Activity modelling in a collaborative ceramic tile design chain: an enhanced IDEF0 approach. Research in Engineering Design, 19(1), 1-20.

Rouncefield, M. (2011). Fieldwork, Ethnography and Ethnomethodology In I. Sommerville (Ed.), LSCITS Socio-Technical Systems Engineering Handbook. St Andrews: University of St Andrews.

Ruinan, G., Qing, L., Xin, L., & Qing, W. (2004, 10-13 Oct. 2004). Modelling for business process design: a methodology based on causal loop diagram. Paper presented at the Systems, Man and Cybernetics, 2004 IEEE International Conference on.

Solomon, L. (1960). The influence of some types of power relationships and game strategies upon the development of interpersonal trust. The Journal of Abnormal and Social Psychology, 61(2), 223-230.

Sommerville, I. (2007). Causal Responsibility Models. In G. D. John Dobson (Ed.), Responsibility and Dependable Systems (pp. 187-207). London: Spring.

Sommerville, I., Dewsbury, G., Clarke, K., & Rouncefield, M. (2006). Dependability and Trust in Organisational and Domestic Computer Systems In K. Clarke, G. Hardstone, M. Rouncefield & I. Sommerville (Eds.), Trust in Technology: A Socio-Technical Perspective: Springer.

Sommerville, I., Storer, T., & Lock, R. (2009). Responsibility Modelling for Civil Emergency Planning. Risk Management, 11(3-4), 179-207.

Sonnenwald, D. H. (1995). Contested collaboration: a descriptive model of intergroup communication in information system design. Information Process Management, 31(6), 859-877.

Sterman, J. (2000). Business Dynamics: Systems Thinking and Modeling for a Complex World. London: McGraw-Hill Higher Education.

Stewart, J., & Williams, R. (2005). The Wrong Trousers? Beyond the Design Fallacy: Social Learning and the User. In H. Rohracher (Ed.), User Involvement in innovation processes. Strategies and limitations from a socio-technical perspective. Munich.

Strens, R., & Dobson, J. (1993). How responsibility modelling leads to security requirements. Paper presented at the Proceedings on the 1992-1993 workshop on New security paradigms.

Suchman, L. (1987). Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge: Cambridge University Press.

Sundström, G. A., & Hollnagel, E. (2008). Modelling Risk in Financial Services Systems: A Functional Risk Modelling Perspective. Paper presented at the Third resilience engineering symposium.

Sutcliffe, A. (2011). Analyzing the Effectiveness of Human Activity Systems with I*. In E. Yu, P. Giorgini, N. Maiden & J. Mylopoulos (Eds.), Social Modeling for Requirements Engineering. Cambridge: MIT Press.

Sutrisna, M., & Barrett, P. (2007). Applying rich picture diagrams to model case studies of construction projects. Engineering, Construction and Architectural Management, 14(2), 164-179.

Telford, B., Cropper, S., & Ackermann, F. (1992). Quality Assurance and Improvement: The Role of Strategy Making. International Journal of Health Care Quality Assurance, 5(3), 6.

Tjosvold, D., Poon, M., & Yu, Z.-y. (2005). Team effectiveness in China: Cooperative conflict for relationship building. Human Relations, 58(3), 341-367.

Trist, E. (1981). The evolution of socio-technical systems.Unpublished manuscript, Toronto.

Vázquez, A., Oliveira, J. G., & Barabási1, A.-L. (2005). Inhomogeneous evolution of subgraphs and cycles in complex networks. Physical Review E, 71(2), 025103.

Venkatesh, V., Morris, M., Davis, G., & Davis, F. (2003). User Acceptance of Information Technology: Toward a Unified View. MIS Quarterly, 27(3).

Vespignani, A. (2012). Modelling dynamical processes in complex socio-technical systems. [10.1038/nphys2160]. Nat Phys, 8(1), 32-39.

Vidgen, R. (1997). Stakeholders, soft systems and technology: separation and mediation in the analysis of information system requirements. Information Systems Journal, 7(1), 21-46.

Viller, S., & Sommerville, I. (1999). Coherence: An Approach to Representing Ethnographic Analyses in Systems Design. Human-Computer Interaction, 14, 9-41.

Viller, S., & Sommerville, I. (2000). Ethnographically informed analysis for software engineers. International Journal of Human-Computer Studies, 53(1), 169-196.

Waring, T., & Wainwright, D. (2002). Communicating the complexity of computer-integrated operations: An innovative use of process modelling in a North East hospital Trust. International Journal of Operations & Production Management, 22(4), 394 - 411.

Wastell, D. (2001). Barriers to effective knowledge management: Action research meets grounded theory. Journal of Systems and Information Technology, 5(2), 21-36.

White, D., & Johansen, U. (2005). Network Analysis And Ethnographic Problems: Process Models Of A Turkish Nomad Clan. Maryland: Lexington Books.

Williams, R., Stewart, J., & Slack, R. (2005). Social Learning in Technological Innovation: Experimenting with Information and Communication Technologies. Cheltenham: Edward Elgar Publishing.

Williams, T., Ackermann, F., & Eden, C. (2003). Structuring a delay and disruption claim: An application of cause-mapping and system dynamics. European Journal of Operational Research, 148(1), 192-204.

Winner, L. (1997). Technology as Forms of Life. In K. Shrader-Frechette & L. Westra (Eds.), Technology And Values. Oxford: Rowman & Littlefield Publishers.

Woltjer, R., & Hollnagel, E. (2008). Functional modeling for risk assessment of automation in a changing air traffic management environment. Paper presented at the 4th International Conference Working on Safety.

Wong, B. (2005). Understanding Stakeholder Values as a Means of Dealing with Stakeholder Conflicts. Software Quality Control, 13(4), 429-445.

Xia, W., & Lee, G. (2004). Grasping the Complexity of IS Development Projects. Communications of the ACM, 47(5), 6.

Yu, E. (2002). Agent-Oriented Modelling: Software versus the World Agent-Oriented Software Engineering II. In M. Wooldridge, G. Weiß & P. Ciancarini (Eds.), (Vol. 2222, pp. 206-225): Springer Berlin / Heidelberg.

Yu, E., Giorgini, P., Maiden, N., & Mylopoulos, J. (2011). Social Modeling for Requirements Engineering. Cambridge: MIT Press.

Yu, E., & Mylopoulos, J. (1994). From E-R to "A-R" - Modelling Strategic Actor Relationships for Business Process Reengineering. Paper presented at the Proceedings of the13th International Conference on the Entity-Relationship Approach.

Yu, E. S. K. (1997). Towards modelling and reasoning support for early-phase requirements engineering. Paper presented at the Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on.

Zhang, J., Smith, R., & Watson, R. B. (1997). Towards computer support of the soft systems methodology: an evaluation of the functionality and usability of an SSM toolkit. European Journal of Information Systems, 6(2), 10.

Zhang, X., Dhaliwal, J. S., Gillenson, M. L., & Moeller, G. (2008). Sources of Conflict between Developers and Testers in Software Development: A

Preliminary Investigation. Paper presented at the Americas Conference on Information Systems 2008, Toronto.

Appendix A – Example Conflict Analysis Template

LASCAD Manager	Analysis
Responsibility 1. Identifies changes to *what* tasks an actor is to perform.	*Manage operational performance of London Ambulance Dispatch* -Ensure that operational performance meets/exceeds ORCON requirements. *Forecast future operational resource requirement of London Ambulance Service* - Introduction of new task
Responsibility 1. Identifies potential changes in *how* the actor(s) will perform tasks	*Manage operational performance of London Ambulance Dispatch* Old 1. Previous manual system did not provide rigorous performance metrics New 1. New IT system provides data to management for analysis. E.g. call answering times, response percentages within 14 minutes.
Time, resources, and capabilities) 1. Identifies any changes in individuals required time, resource or capabilities ensure appropriate fulfilment of tasks 2. Identifies potential risks of not having required time, resources or capabilities	*Time* **There is a risk that LAS Management may resist the IT system as the perceive it to degrade their chances of meeting ORCON targets.** *Resources* **There is a risk that LAS Management may resist the IT system if they perceive they will be given inadequate resources to for it to function.** **Risk Mitigated by top-down pressure to meet ORCON targets.** *Capabilities* **There is a risk that LAS Management will resist the system if they do not or are not willing to develop skills to manage the system.** **Risk Mitigated by top-down pressure to meet ORCON targets.**
Values, status & satisfaction 1. Identify agents values and satisfaction 2. Identify risks of Incompatibility between required activity and individual values and satisfaction criteria	*Values* 'providing ambulance services at or above national requirements' **There is a risk that LAS Management may resist the IT system if they perceive the system to be a threat to meeting requirements - due to dependency on new technology (perhaps unreliable) and untested processes to meet targets. Risk Mitigated since radical technology perceived as the only way to meet targets.** *Status* Benefit: LAS Management may perceive the IT system as increasing their status as would be most technologically advanced in country. *Satisfaction* 'providing ambulance services at or above national requirements' **There is a risk that LAS Management may resist the IT system if they perceive the system to be a threat to meeting requirements - due to dependency on new technology (perhaps unreliable) to meet targets.** **Risk Mitigated since radical technology perceived as the only way to meet targets**.
Multiple incompatible	None Noted.

responsibilities 1. Identify risks that an individual is assigned multiple responsibilities with incompatible activities or assessment metrics	
Relational (Incompatibility between actors)	*Control Room Assistants & Ambulance Crew* Ongoing pay disputes between NUPE and LAS resulting in tension and mistrust between LAS management. **There is a risk that LAS Management may view Control Room assistant or ambulance crew as being obstructive when providing negative feedback about the system due to poor relations.** *LAS Executives* Implementation follows a recent downsizing of LAS Management resulting in a continuing fear for jobs. **There is a risk that LAS Management may be reluctant to report negative information to executives for fear of losing their jobs due to recent downsizing by LAS Executives**
Relational (Procedural / Distributive Injustice)	*Procedural Injustice* **There is a risk that the LAS Managers may perceive that the imposition of the IT system by LAS executives (due to ORCON requirements) is unjust because LAS is an exceptional case due to the London area being a far larger region than any other. (ORCON requirements should not apply)** *Distributive Injustice* **There is a risk that LAS Managers may perceive that the imposition of the IT system by LAS executives (due to ORCON requirements) is distributively unjust as other ambulance services will have to make less changes to meet targets due to the smaller scale of other regions.**

Appendix B – Risk Tables for LASCAD92/96

LAS Manager	
Risks	
There is a risk that LAS Management may resist the IT system as they perceive it to degrade their chances of meeting ORCON targets. 92&96Risk Mitigated as management perceive technology as the only approach that will enable the meeting of ORCON targets.	M1
There is a risk that LAS Management may resist the IT system if they perceive they will be given inadequate time/resources to for it to function. 92Risk Inappropriately Mitigated as LAS Management fearful for their jobs and under intense pressure to meet deadlines. 96Risk Mitigated by perception of top-down commitment to provide what ever is required for success. E.g. Investment in new control room, additional staff.	M2
There is a risk that LAS Management will resist the system if they cannot or are not willing to develop skills to manage the system or its development. 92Risk Inappropriately Mitigated by top-down pressure to meet ORCON targets and risk of job loss. 96Risk Mitigated by top-down pressure to meet ORCON targets, acquisition of additional resources and restructuring to bolster management and operational staff.	M3
There is a risk that LAS Management may resist the IT system if they perceive the system to be a threat to meeting requirements - due to dependency on new technology (perhaps unreliable) and untested processes to meet targets. 92&96Risk Mitigated since radical technology perceived as the only way to meet targets.	M4
There is a risk that LAS Management may be reluctant to report negative information to executives for fear of losing their jobs due to recent downsizing by LAS Executives 96Risk Partially Mitigated since flexible time-frame adopted removing pressure for immediate results and recent hiring of management and operational staff suggesting jobs not at risk.	M5
There is a risk that LAS Management may view Control Room assistants or ambulance crew as being obstructive when providing negative feedback about the system due to poor past relations.	M6
There is a risk that the LAS Managers will resist as they perceive that the imposition of the IT system by LAS executives (due to ORCON requirements) as procedurally unjust because LAS is an exceptional case due to the London area being a far larger region than any other. (ORCON requirements should not apply) 92Risk Inappropriately Mitigated as LAS Management fearful for their jobs and under intense pressure to meet deadlines. 96Risk Partially Mitigated since flexible time-frame adopted allowing LAS additional time to meeting the ORCON requirements.	M7
There is a risk that LAS Managers will resist as they perceive that the imposition of the IT system by LAS executives (due to ORCON requirements) is distributively unjust as other ambulance services will have to make less changes to meet targets due to the smaller scale of other regions. 92Risk Inappropriately Mitigated as LAS Management fearful for their jobs and under intense pressure to meet deadlines. 96Risk Partially Mitigated since flexible time-frame adopted allowing LAS additional time to meeting the ORCON requirements.	M8

Control Room Assistant	
Risks	
There is a risk that Control centre assistants will resist change because of a perception of job cuts because many of their existing tasks will be computerised. 96Risk Partially Mitigated since additional operational staff hired	C1
There is a risk that Control centre assistants will resist the system because they perceive they will not be given adequate resources to perform their tasks using the system. 96Risk Partially Mitigated since control room upgraded with new electrical systems and digital phones.	C2

There is a risk that Control centre assistant will resist the change because of a lack of capabilities to operate computerised systems. 96Risk Mitigated since new software developed using a participative approach whereby software only goes live once users are confident in the capabilities of the release and their ability to use it.	C3
There is a risk that Control centre assistants may perceive the changes in processes as degrading their ability to meet their values of rapid patient as now dependent upon the function of an IT system 96Risk Mitigated since new software developed using a participative approach whereby software only goes live once users are confident in the capabilities of the release its compatibility with their values.	C4
There is a risk that Control centre assistant perceive the IT system as trivialising/routinizing the work thus reducing its status. 96Risk Mitigated since new software supports/enhances control room assistants decision-making rather than taking over their decision-making. E.g. Rather than automatically selecting resources the system provides the operator with suggestions for suitable resources and their locations.	C5
There is a risk that Control centre assistant perceive the IT system as removing satisfying work as it automates important decisions such as resource allocation which employees may have pride in performing. 96Risk Mitigated since new software supports/enhances control room assistants decision-making rather than taking over their decision-making.	C6
There is a risk that Control Room assistants will perceive the project negatively if it is perceived as a LAS Management/LAS Executive initiative due to a long standing history of disputes with LAS Management & LAS Executives regarding pay and also the failed introduction of previous systems. 96Risk Partially Mitigated since significant changes to LAS Management / Executives were made and also their actions (such as hiring additional staff and providing new equipment) suggest a management style based upon cooperation and gradual change as approved by staff.	C7
There is a risk that Control Room assistants may perceive the project as distributively unfair as they face major changes and perhaps job losses and in return they receive little benefit. 96Risk Partially Mitigated since control room assistants receive an improved control room which is more comfortable, are more in control of the changes made and the atmosphere of job losses is minimal due to recent hiring.	C8

Ambulance Crew	
Risks	
There is a risk that without adequate training the Ambulance crew will be unable to operate the equipment or lack trust in equipment 96Risk Mitigated as ambulance crew involved in testing and approving equipment prior to go-live.	A1
There is a risk that the IT system will interfere with values of crew (rapid patient care) as they must follow instructions of machine even if obviously suboptimal. 96Risk Partially Mitigated as ambulance crew involved in testing and approving equipment prior to go-live.	A2
There is a risk that the IT system will interfere with values of crew if it does not facilitate the taking into of crew experience and local knowledge. 96Risk Partially Mitigated as ambulance crew involved in testing and approving equipment prior to go-live.	A3
There is a risk that Ambulance Crew may perceive the system to reducing the status of their work as it automates their decision-making process such that they must follow instructions from screen only. 96Risk Partially Mitigated as ambulance crew involved in testing and approving equipment prior to go-live.	A4
There is a risk that the IT system will interfere with satisfaction of crew as does not give them autonomy to use their crew experience and local knowledge for which they have pride. 96Risk Partially Mitigated as ambulance crew involved in testing and approving equipment prior to go-live.	A5
There is a risk that Ambulance crews will perceive changes to processes brought about by the IT system as negative (interference) because of ongoing problems with staff consultation 96Risk Mitigated as ambulance crew involved in testing and approving equipment prior to go-live.	A6
There is a risk that Ambulance crews will perceive the process as procedurally unjust due to their lack of involvement in consultation. 96Risk Partially Mitigated as ambulance crew involved in testing and approving equipment prior to go-live.	A7
There is a risk that Ambulance crews will perceive the changes as distributively unjust due to losing a large chunk of autonomy for which they derive no/little benefit. 96Risk Partially Mitigated as ambulance crews received upgraded ambulances that makes work more comfortable and were issued with personal radios so they can communicate between themselves.	A8

Appendix C – Mapping of Identified Causes and Conflict Analysis Risks

Table C.1 – Identified causes of failure and identified conflict derived risk

Identified cause of Failure	Identified conflict derived risk
Executives did not appreciate that LASCAD was a business re-engineering project (P. Beynon-Davies, 1995) (Hougham, 1996)	*Executives outside of scope of conflict analysis performed as little information obtainable from literature.*
Due to external pressure to achieve results, insufficient time was allowed for development and testing of the extremely complex technical solution (Page, 1993) (P. Beynon-Davies, 1995)(Hougham, 1996) (Beynon-Davis, 1999)	The underlying source of this failure was the inappropriate mitigation of risk [M2] using coercion rather than provision of time and resources.
There was a lack of disciplined technical approach (Page, 1993) (Finklestein, Dowell, 1996)(Hougham, 1996) (Beynon-Davis, 1999)	The underlying source of this failure was the inappropriate mitigation of risk [M3] using coercion rather than ensuring management had appropriate skills and knowledge.
There was little attempt to manage the changes in the organisations culture as part of the project (Hougham, 1996)	The underlying source of this failure was the inappropriate mitigation of risk [M3] using coercion rather than ensuring management had appropriate skills and knowledge.
Poor user involvement, lack of ownership and some evidence of resistance (Page, 1993) (P. Beynon-Davies, 1995)(Finklestein, Dowell, 1996) (Beynon-Davis, 1999)	The underlying source of this failure was the lack of mitigation of risk [M6] resulting in management ignoring stakeholder feedback.
Evidence of irrational persistence in relation to continuing to use an approach with a tightly constrained budget and time-scale (Page, 1993)(Beynon-Davis, 1999)	The underlying source of this failure was the inappropriate mitigation of risk [M3] using coercion resulting in management's irrational persistence.
Developers had little experience of developing critical systems [(Page, 1993)(P. Beynon-Davies, 1995)	The underlying source of this failure was the inappropriate mitigation of risk [M3] using coercion resulting in management being able to admit they lacked the necessary tendering and acquisition skills or spending time to remediate this.
History of failure (P. Beynon-Davies, 1995)(Finklestein, Dowell, 1996)	The underlying source of this failure was the lack of mitigation of risk [C7] that resulted in the perception that the project was a management power play.
No formal Project Management Methodology followed e.g. PRINCE (Page, 1993)(P. Beynon-Davies, 1995)(Finklestein, Dowell, 1996)	The underlying source of the failure was the inappropriate mitigation of risk [M3] via coercion rather than enabling/ensuring management had the appropriate skills and knowledge to manage the project.
Fear of reporting failure & Misleading LAS Executives over experience of Contractors (Page, 1993)(P. Beynon-Davies, 1995)	The underlying source of the failure was the lack of mitigation of risk [M5] resulting in fear of reporting failure.
Lack of adequate testing (Page,	The underlying source of the failure was the inappropriate

1993)(P. Beynon-Davies, 1995) (Finklestein, Dowell, 1996)	mitigation of risks [M2] [M3] using coercion. Management were given neither the time, the resources, nor the skills to ensure inadequacies in vendors' practices could be identified and addressed.
Training was limited (Page, 1993)(P. Beynon-Davies, 1995) (Finklestein, Dowell, 1996)	The underlying source of the failure was caused by the inappropriate mitigation of risk [M2] via coercion rather provision of time/resources.
Communication and response time problems (Page, 1993)(Beynon-Davis, 1995)	The underlying source of the failure was caused by the inappropriate mitigation of risk [M2] via coercion rather provision of time/resources and the lack of mitigation of risk [M5].
Frustration of Ambulance crews (Page, 1993)(Beynon-Davis, 1995)	The underlying source of the failure was the lack of mitigation of risks [A1] [A2] [A3] [A5]
Anti-computer bias (Beynon-Davis, 1995)	The underlying source of the failure was the lack of mitigation of risks [C5] [A4]
Poor NHS & Labour relations background & Low moral (Page, 1993)(Beynon-Davis, 1995) (Finklestein, Dowell, 1996)	The underlying source of the failure was the lack of mitigation of risks [C7], [C8], [A6], [A8]
Lack of strategic vision (Page, 1993)(Beynon-Davis, 1995)	*Executives outside of scope of SIA performed as little information obtainable from literature.*
Aggressive pace of change (Page, 1993)(Beynon-Davis, 1995)	The underlying source of the failure was the lack of mitigation of risk [A7] and the inappropriate mitigation of risk [M2] resulting in an aggressive pace of change.
Lack of investment in LAS (Page, 1993)(Beynon-Davis, 1995)	The underlying source of the failure was the lack of mitigation of risks [C7] and [A6] resulting in a lack of investment.
Evidence of ignoring outside advice on tightness of timetable or high risk of system (Page, 1993) (Beynon-Davies, 1995)	The underling source of the failure was the lack of mitigation of risk [M5] and the inappropriate mitigation of [M2] and [M3].
Tendering process focused on price over quality (Page, 1993)	The underlying source of the failure was the inappropriate mitigation of risks [M2] [M3] resulting inappropriate tendering

Appendix D - Risk Tables for Oil/Gas Case Study

Risks from Support Manager Perspective

Support Manager	
Risks	
There is a risk that Support manager will conflict with the change because in situations where external providers are not forthcoming in resolving support requests managerial time may be significantly diverted.	SM1
There is a risk that the Support Manager will conflict with the change because they perceive it will result in a downsizing of their department. This may occur because work may be perceived to require fewer resources (Systems engineers) since network/hardware support and back-ups is provided by external provider.	SM2
There is a risk that the Support manager may require additional resources (Systems Engineers) if staff experience & knowledge of 'EC2' administration results in slower work in comparison to local environment.	SM3
There is a risk that Support manager will conflict with the change because they perceive the 'EC2' style provision to interfere with provision of SLA. This could occur because support is dependent upon: i) external provider that is out of their direct control. ii) it comprises a new technology that staff have little experience with. iii) There are additional uncertainties which impact service quality that is out of their direct control such as network connection between external service provider and customer.	SM4

Risks from Support Engineer Perspective

Support Engineer	
Risks	
There is a risk that support engineers will conflict with the change because they may perceive it interferes with their role with respect to h/w, network and back-up support activities as a reduction in workload and thus a threat to their jobs.	SE1
There is a risk that support engineers will conflict with the change because they may perceive their interactions with external providers as time consuming and un-enjoyable and a threat of being overworked	SE2
There is a risk that support engineers will conflict with the change because they perceive the change as the creation of additional unsatisfying work (the need to acquire new skills).	SE3
There is a risk that support engineers will not have enough experience & requisite knowledge of EC2 administration resulting in rework & increased time/effort to accomplish tasks.	SE4
There is a risk that support engineers will conflict with the change because changes result in employees being less hands-on with technology at a hardware, network and back-up level thus reducing their satisfaction.	SE5
There is a risk that support engineers will conflict with the change as reliance on external provider introduces dependence (and thus an uncontrollable uncertainty) to being perceived to be good at system support thus reducing certainty that they will be satisfied by their work.	SE6

Risks from Sales / Marketing Perspective

Sales / Marketing	
Risks	
There is a risk that sales/marketing will conflict with the change if they are not provided with suitable promotional resources.	S1
There is a risk that sales/marketing staff will conflict with the change because if they do not understand the benefits of the product to customers.	S2
There is a risk that sales/marketing will conflict with the change because if sales/marketing persons are set unrealistic goals with regards to selling the 'EC2' style offerings as this would compromise their satisfaction.	S3

Risks from Finance / Business Perspective

Finance / Business Development	
Risks	
There is a risk that finance/business development staff will conflict with the change because calculating profits over a long-term period will become more uncertain unless costs are transferred directly to customer.	F1
There is a risk that finance staff will conflict with the change because profit calculations become more time consuming. This can occur if external provider charges are not directly passed to customer. This is because costs associated with consumption of bandwidth and processing power vary with extent of usage and may change with market forces.	F2
There is a risk that financial staff will conflict with the change because 'EC2' is perceived to decrease the financial soundness of the organisation by opening it to significant fluctuations in bandwidth and processing costs.	F3

Risks from Customer Relations Perspective

Customer Relations Staff	
Risks	
Description	No
There is a risk of deterioration in customer care as in some situations it may take longer to resolve customer queries as cooperation with external service provider may be necessary. This may result in a backlog of work resulting in additional pressure being placed on staff.	CR1
There is a risk of deterioration in customer service quality if Customer relations staff are not given appropriate knowledge of 'EC2' style services. This may occur as effort would be wasted resulting in rework and wasted customer time resulting in a deterioration of customer service quality.	CR2
There is a risk that customer relationship staff will conflict with the change because the change may be perceived to compromise their satisfaction of providing fast and accurate responses to customers' queries. The 'EC2' style deployment will create additional dependencies (with an external service provider) resulting in customer care being out of their direct control in situations related to h/w, network connectivity, and availability.	CR3

Appendix E – Survey Questions

LSCITS
Large Scale Complex IT Systems

Survey: Electronic Document Management Usage & Issues

Introduction
We are investigating the use of document management systems and the impact they have on people’s work as part of a UK Strategic Initiative called LSCITS.

Purpose
The aim of this research is to understand peoples’ experiences of using document management systems and the impact they have on work. To address a need to develop cost-effective approaches that help practitioners understand and manage socio-complexity.

Procedure
We would be grateful if you could spend ten minutes of your time to share with us your views by completing this questionnaire. You may feel there is an element of repetition in some of the questions - this is necessary to ensure construct validity and is minimized as much as possible.

Thanks,

David Greenwood dsg22@cs.st-andrews.ac.uk
PhD Student – Computer Science – University of St Andrews

Questionnaire Begins

1. Please provide a short introduction to your role within <organisation>?

2. Please describe your responsibilities?

#	Description	
1		
2		
3		
4		
5		
6		
7		

3. What are your day-to-day activities?

#	Description	**Proportion of time**
1		
2		
3		
4		
5		
6		
7		

4. What are your most serious work challenges/problems?

#	Description	
1		
2		
3		

4		
5		
6		
7		

5. Please describe your History with <SYSTEM>?

6. Which of your activities and responsibilities does <SYSTEM> help you accomplish

8. How does <SYSTEM> help you accomplish your activities and responsibilities

9. What problems does using <SYSTEM> introduce to your work – how do you work around them?

10. What aspects of <SYSTEM> impede your accomplishment of activities and responsibility?

Organisational Complexity

1. Continuing investment, development and maintenance of <SYSTEM> ______________________ my interests, goals, responsibilities, or values

Strongly interferes with			Neither interfere with nor facilitate			Strongly facilitates

2. <SYSTEM> affects my **working practices** in ways that ______________________ my interests, goals, responsibilities, or values

Strongly interfere with			Neither interfere with nor facilitate			Strongly facilitate

3. <SYSTEM> affects my **time** in ways that ______________________ my interests, goals, responsibilities, or values

Strongly interfere with			Neither interfere with nor facilitate			Strongly facilitate

4. <SYSTEM> affects my ability to acquire **sufficient or appropriate resources** in ways that ________________________ my interests, goals, responsibilities, or values

Strongly interfere with			Neither interfere with nor facilitate			Strongly facilitate

5. <SYSTEM> **affects** my **capabilities or skills** in ways that ________________________ my interests, goals, responsibilities, or values

Strongly interfere with			Neither interfere with nor facilitate			Strongly facilitate

6. <SYSTEM> is _________ with my **values** e.g. privacy, confidentiality, risk aversion

Strongly Incompatible			Neither Compatible / Incompatible			Strongly Compatible

7. <SYSTEM> _________ **affects** my **job satisfaction**

Adversely			Neither favorably / adversely			Favorably

8. <SYSTEM> _________ **affects my status**

Adversely			Neither favorably / adversely			Favorably

9. <SYSTEM> causes me to **cooperate with people/groups** where in the **past there have been tensions or rivalries**

Strongly Disagree			Neither agree/disagree			Strongly Agree

10. The distribution of project **benefits and drawbacks** is ______ to me

Unfair			Neither fair / unfair			Fair

11. For me the **benefits** of <SYSTEM> are ________ its **drawbacks**

Smaller than			Equal to			Greater than

12. To what extent are you **satisfied** with <SYSTEM>

Strongly Dissatisfied	Dissatisfied	Slightly Dissatisfied	Neither satisfied/ dissatisfied	Slightly Satisfied	Satisfied	Strongly Satisfied

13. To what extent are you supportive of continuing investment, development and maintenance of <SYSTEM>

Strongly unsupportive			Neither unsupportive or supportive			Strongly Supportive

14. To what extent is <SYSTEM> important to your interests, responsibilities, goal and values

Very unimportant			Neither important or unimportant			Very Important

15. To what extent do you feel ownership of <SYSTEM>

None						Strong

Efficiency & Effectiveness

1. <SYSTEM> affects **my speed at accomplishing activities**

Significantly slower	Slower	Slightly slower	Neither faster nor slower	Slightly faster	Faster	Significantly faster

2. <SYSTEM> affects **my productivity**

Significantly worsen	Worsen	Slightly worsen	Neither improve nor worsen	Slightly Improve	Improve	Strongly Improve

3. <SYSTEM> **takes little effort to use on my part**

Strongly Disagree	Disagree	Slightly Disagree	Neither agree/disagree	Slightly Agree	Agree	Strongly Agree

4. <SYSTEM> **creates me unnecessary work** and thus **wastes effort**

Strongly Disagree	Disagree	Slightly Disagree	Neither agree/disagree	Slightly Agree	Agree	Strongly Agree

5. To make <SYSTEM> **fit the way I work** I have to use **work-arounds**

Strongly Disagree	Disagree	Slightly Disagree	Neither agree/disagree	Slightly Agree	Agree	Strongly Agree

Information Quality

1. The accuracy of information in <SYSTEM> is typically

Problematic			Neither problematic or meets my needs			Meets my needs

2. The timeliness of information in <SYSTEM> is typically

Problematic			Neither problematic or meets my needs			Meets my needs

3. The completeness of information in <SYSTEM> is typically

Problematic			Neither problematic or meets my needs			Meets my needs

4. The relevance of information in <SYSTEM> is typically

Problematic			Neither problematic or meets my needs			Meets my needs

5. The consistency of information in <SYSTEM> is typically

Problematic			Neither problematic or meets my needs			Meets my needs

System Quality

1. The reliability and uptime of <SYSTEM> is

Problematic			Neither problematic or meets my			Meets my needs

			needs			

2. The user interface of <SYSTEM> is

Problematic			Neither problematic or meets my needs			Meets my needs

3. The system features for finding relevant documents are

Problematic			Neither problematic or meets my needs			Meet my needs

4. Ease of establishing accuracy and timeliness of documents

Problematic			Neither problematic or meets my needs			Meets my needs

Document Management Policy

1. There is limited Understanding of policies and procedures

Strongly Disagree						Strongly Agree

2. There is uncertainty of **what** and **Where** content should be stored

Strongly Disagree						Strongly Agree

Support Service Quality

1. Extent and timeliness of training provided

Inadequate						Adequate

2. Extent of your understanding of how <SYSTEM> can help you accomplish your responsibilities and activities

Insufficient						Sufficient

3. To what extent are you **informed** about how <SYSTEM> can help accomplish your work

Very uninformed	Uninformed	Slightly uninformed	Neither informed nor uninformed	Slightly informed	Informed	Very informed

4. Extent of your feeling of involvement/participation with <SYSTEM> deployment

Insufficient						Sufficient

5. The processing of change requests is _______

Slow						Fast

6. Time required for development of changes to <SYSTEM> is __________

Unreasonable						Reasonable

5. Relationship with <SYSTEM> IT Staff is _____

Dissonant						Harmonious

6. Attitude of <SYSTEM> IT Staff is ________

Belligerent						Cooperative

7. Communication with <SYSTEM> IT Staff is ______________

Destructive						Productive

8. The Quality of communication of <SYSTEM> IT staff

Problematic						Excellent

9. To what extent do you trust <SYSTEM> IT staff

None						Strong

10. In the past the relationship between users and <SYSTEM> IT staff has been

Conflictual						Cooperative

END OF QUESTIONNAIRE

Please feel free to write comments in the space below and overleaf if necessary: