Understanding Complexity in Architecture-based Analysis

David Colquitt (BE Honours)

A thesis submitted for the degree of
Doctor of Philosophy

University of Technology, Sydney
Faculty of Engineering (Information and Communication Technology Group)

2007

CERTIFICATE

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Production Note:
Signature removed prior to publication.

ACKNOWLEDGEMENTS

A research degree by thesis is a huge undertaking and affects your support network as much as it does you. The continued support and encouragement of the following people have made this research possible. This is a small gesture of thanks for their often overwhelming efforts.

Firstly I must thank my principal supervisor Professor John Leaney, whose ability to balance professional guidance with human compassion made him the perfect mentor. He gave both his time and energy when he could least afford to, for this I am eternally indebted.

To my co-supervisor Dr Tim O'Neill, whose drive and energy I constantly drew upon. Our swimming sessions helped keep me sane and showed me how important a work/life balance is.

To my wife, thank you for supporting me on this journey. You bore as much of the highs, lows and prolonged anxiety as anyone, yet only returned love and understanding. My love for you will always ground my worry over other comparatively trivial pursuits.

Likewise, thank you to my friends who have shown me that human relationships are the most rewarding and long lasting endeavour one could undertake. In particular to Mark, Toby, Pete, Ange, Cara and Kristen, whose continued encouragement meant the world to me.

To my mum, dad, brother and sister, thank you for providing the support platform, from which I feel I can fearlessly pursue life. You have instilled in me the desire to follow my heart and the passion to see it through.

Finally I would like to thank Alcatel Australia and the Australian Research Council for their financial support, as well as the faculty of Engineering, Institute for Information & Communications Technologies (ICT) and graduate school for their continued support of my candidature.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ACRONYMS

ABD	Architecture-Based Design
ADL	Architecture Description Language
AQA	Architecture-based Quality Assessment
AR	Action Research
ASD	Architecturally Significant Decision
ASR	Architecturally Significant Requirement
ATAM	Architecture Tradeoff Analysis Method
GBR	Goal Based Requirements
HAS	Human Activity System
IS	Information Systems
MEEN	Management of Enriched Experience Networks
NGN	Next-Generation Network
OSS	Operational Support Systems
QDS	Quantified Design Space
QOE	Quality of Experience
RMA	Rank Matrix Analysis
SA	Software Architecture
SAABNET	Software Architecture Assessment using Bayesian Networks
SAAM	Software Architecture Assessment Method
SARA	Software Architecture Review Assessment
TMF	Telecommunications Management Forum
TMN	Telecommunications Management Network
UTS	University of Technology Sydney

ABSTRACT

As computer-based systems become more embedded in organisations and integrated into organisational activity, they also become proportionately more complex. Telecommunications networks in particular are facing significant challenges as their infrastructure combines to form one of the largest, most heterogenous systems around. The increase in complexity, coupled with the cost of late changes to system designs, elevates the importance of being able to reason about system designs from the earliest artefacts onwards. Software architecture is a discipline designed to address the increase in complexity by facilitating early design reasoning and providing a complimentary focus on system quality as well as function.

The following thesis reports on a research project aimed at addressing the complexity of the telecommunications design task with the techniques of software architecture. A particular focus is given to architecture-based analysis, the motivation for which arose from reconciling experience in design meetings against the focus of the analysis methods. Combining this experience with a diverse examination of systems literature realised the 'hypothesis' that the existing analysis literature did not address the true complexity of the task. Using a collaborative design project as a platform, the research made use of the situated method of inquiry called action research to explore the complexity of the analysis task.

The learning outcomes present the manifestations of complexity observed in the ATAM process in terms of a people and systems dimension. These aspects of complexity are shown to affect some of the most important ATAM objectives. Insight is also offered on the use of the method with respect to the design lifecycle, discussing how the elements of the design situation and situational complexity conspire to diffuse the efficacy of the ATAM. Some future resolution to this is suggested in terms of splitting out the analysis objectives and maintaining two streams of analysis, as well as paying attention to the content aspects of the process that drive its direction from within. While all the individual learning outcomes are important, the most enduring outcome stems from the rich understanding obtained by entertaining a 'soft' perspective of the analysis task. This is perhaps no better summed up than by Bucciarelli.

"attempts to improve the engineering design process by critics and assessors of that process have been, for the most part, couched wholly in instrumental terms... these instrumental approaches are deficient when applied to design process considered as a social process awash in uncertainty and ambiguity. They miss many of the trees in the forest." (Bucciarelli, 2002, pp 221)

1

Introducing the research

Introduction

The following thesis examines the design of computer-based systems (CBS) with a particular focus on expounding the complexity of performing architecture-based (design) analysis from an early point in the systems lifecycle onwards. A theoretical framework in the spirit of Checkland (Checkland & Holwell, 1998) is developed in order to define complexity in the context of architecture-based analysis. Learning with respect to the framework is recorded as the action research methodology is used to investigate architecture-based analysis within a collaborative industry-academic research project.

Chapter 1 introduces the thesis, outlining the motivation for undertaking the research and its importance within the context of existing knowledge in the area. The research situation and methodology are presented, in light of which the research objectives are put forward. The structure of the thesis is depicted and a brief synopsis of each chapter given. Finally the original contributions of this work are identified and any publications stemming from the work acknowledged.

1.1. Motivation for this thesis (The research issue)

1.1.1. Systems quality and software architecture

The design of computer-based systems is seen as a challenging and often complex task, fraught with danger in many ways. Historical evidence of the difficulty in designing computer-based systems is common within the industry and academic reporting. This historical evidence ranges from statistical studies of system failures (The-Standish-Group, 1994), to specific instance of failure (Lindberg, 1999), to retrospectives and pragmatic aphorisms of well respected veterans of the field (Brooks, 1987). In many instances the failure represents the neglect of an aspect of the system that is critical to the expectations of the client. Costly and even fatal lessons have demonstrated that customer expectations extend beyond the function of the system into what have since been generically termed quality attributes (IEEE, 1998c).

Software systems have traditionally been viewed in a functional capacity, consequently design and testing activity focused on the specific function of software in its domain of use (Bass et al., 1998) (Bosch & Molin, 1999). Yet as Len Bass[1] once pointed out in a public lecture, systems are rarely judged as failures on the basis of not performing the function expected of them. Instead the disappointment arises from the expectations of the client with regard to the system's quality attributes. For example the system is not robust enough and fails under the slightest environmental changes; doesn't perform as expected and handle large workloads; or is difficult to maintain and extend. These quality expectations are often only implicit in the intentions of the client and need to be made explicit in the requirements and clearly addressed in the design.

Significant work has since been conducted to understand the quality aspects of systems and how they relate to the structure of the system. The knowledge that now exists about quality attributes, how they can be identified, measured and realised, has posited a highly significant relationship between the goals of a system (be they functional or quality-related) and the structure of the system. Importantly, it suggests there is a capacity to design for qualities like performance, maintainability or security through architecture-based decisions. The work to place this knowledge within a set of

[1] http://www.sei.cmu.edu/staff/ljb/ at UniNSW on 13/12/2002 attended by the researcher

methods and techniques that can be practised by system designers has resulted in the discipline of software architecture (SA) (Shaw & Garlan, 1996). The name 'software architecture' reflects the perceived similarities with the built environment where the underlying structure (architecture) is prevalent in determining the properties of the end-product (construction). The tools and methods of software architecture are intended to facilitate designers to reason about the structure of a system in an abstract form, free from the constraints of implementation detail, by which stage design decisions are likely to have become institutionalised (Bass et al., 2003).

1.1.2. Research context – telecommunications management

The need to design for quality as well as function is ubiquitous in all instances of technology, but is perhaps no more pressing than in the field of telecommunications. Communication is an essential part of modern business and social existence creating a dependency which exacts stringent expectations of quality from the perspective of end-users. As Amyot 2003 noted the global reach of telecommunications escalates the scale of modern networks, creating systems of great size and diversity.

> *"Telecommunications systems consist of many distributed components...the global telecommunications network has become the largest, distributed network worldwide, and is characterised by a tremendous heterogeneity".* (Amyot & Eberlein, 2003, pp 62)

Computing platforms and hence software systems are pervasive throughout all stages of telecommunications delivery, from the user interface devices, to the infrastructure used to connect and transmit the information. Consequently these computer-based systems are largely responsible for the quality perceptions of the end users. Quite incongruently, telecommunications management systems, which support the operational processes of telecommunications carriers, have traditionally underperformed. Telecommunications management systems are notorious for failing to meet performance expectations, integrating poorly with peer systems and being unable to evolve to incorporate new functions and processes (Hope & Nichols, 2002).

The lack of management system quality hasn't had a strong effect upon the telecommunications product quality as existing infrastructure is strongly engineered for quality. A good example of this is the fixed line telephony network that effectively establishes a dedicated transmission line between the users for the length of the call. In many ways the management system helps establish the call and the engineering configuration takes care of the rest. As a result the poor quality of management systems has been more perceptible to carriers internally than it has been to external users. However a significant change is afoot in the telecommunications sector. A combination of competitive market pressures and the need to tap new revenue sources is mandating carriers converge towards newer infrastructure platforms. The challenge of the newer technology is that engineered resources and 'light touch' management is set to be replaced by dynamically allocated resources and 'active management'. The desire to provide innovative new service sets to consumers, converge service delivery technologies, and manage down to a per-user session level of granularity has created a step-wise increase in the complexity of management systems (Hope & Nichols, 2002). Keeping in mind the already underwhelming track-record of management systems, the task looms as a risky and potentially exorbitantly expensive exercise.

The emphasis on future management capability as enabling telecommunications carrier ambitions highlights a dependency that breaks the traditional operational mould of carriers. The opportunities for the business are set to become largely dependant upon the capability of the management system (Hope & Nichols, 2002). This change is indicative of a general trend of systems becoming more reliant upon software, replacing a traditional emphasis on mechanical and electronic aspects. Software is evolving from a peripheral concern in systems to a core one, critical for the capability of the system (Bosch, 2003). As software becomes a more dominant aspect of systems it additionally assumes greater responsibility for ensuring the quality attributes of the system (Bosch, 2003).

The importance of the management systems to the strategic vision of carriers has created significant interest in developing a management solution. However the boom and bust of the 'dot-com bubble' has developed an increased awareness of investment risk. The poor track record of management systems to date combined with the step-wise increase in complexity present tangible risks. These risks naturally discourage attempts at placing greater responsibility on software systems in the form of new management platforms.

A discipline like software architecture which focuses on correctly developing the quality as well as the functional aspects of a system, potentially has a lot to offer the next-generation network management problem. The difficulty for commercial enterprises is that software architecture is reasonably prolific within research and academic circles, but is yet to offer a set of rigorous tools that the industry at large can 'popularise' (Shaw, 2001). Concepts like design patterns, architecture-based analysis, and to some extent architectural styles are utilised in the commercial world (Shaw, 2001) but they are targeted techniques only covering parts of the design lifecycle. The greater notions of architecture-based design as espoused by literature such as that of the SEI (Bachmann et al., 2000), remain largely unpractised.

The need to develop the principles of software architecture is a strong theme of the Architecture-Based Engineering Group within the University of Technology, Sydney (UTS). In particular this group is focused on commercial applicability and increasing the use of architecture-based principles in real world design situations. The desire to further architecture-based engineering through real world design situations and the applicability of software architecture principles to telecommunications management formed two significant drivers in realising a collaborative research project. The collaboration, initiated by a major international vendor and UTS sought to combine a popular industry archetype termed 'policy based management' (Flegkas et al., 2002) with open-systems principles and architecture-based design methods in order to create a proof-of-concept management system. The admission of the researcher into the project team from a previous position with a commercial telecommunications vendor marked the entry into the research situation.

The motivation for a specific focus on architecture-based analysis arose from an early project need to perform an architecture-based analysis on existing system designs. The perceived problem with architecture-based analysis arose by reconciling the techniques of the available methods against the current experience in the design meetings. Existing analysis methods lacked the need for developing understanding and granting genuine representation to the diversity of stakeholders likely to participate in such an exercise. Problematically the research team exhibited strong diversity from the outset, and matters of design often became dominated by interpretation and the need for understanding amongst the design group. Participation in design meetings indicated the role of the stakeholder in the project, their experience and their areas of interest all contribute to the ways in which they participated. The analysis methods at hand provided very little guidance on how to handle this diversity and its potential effects upon the process outcomes. Consequently, further understanding about the nature of architecture-based design and the characteristics of existing methods was sought.

1.1.3. Research focus – architecture-based analysis

In reaction to the complexity of systems a critical part of design ethos has become iteration and verification. Iteration implies a system is designed in an incremental fashion, with constant analysis used as a means of verifying the latest design moves are consistent with the intended outcomes. Architecture-based design is no exception and architecture analysis performs a pivotal role of both verifying and understanding designs (Bachmann et al., 2000; Bass et al., 2003), supporting the broader trend in design processes, architecture-based analysis has moved away from methods aimed at selecting from a set of candidate architectures towards methods directed at understanding and aiding the design and evolution of a single architecture. Generic checklists and overall numeric scores have given way to scenarios and risk identification.

With the change in focus from candidate selection to design optimisation, architecture-based analysis has moved from the domain of a few technical experts to that of the broader stakeholder community. A community which has grown proportionately to the number of stakeholders whose opinions legitimately shape the form of the design (Bucciarelli, 2002). Consequently, the more recent methods of analysis strive to give representation to all key stakeholder groups. In doing so these methods also raise concerns associated with managing 'the non-technical aspects of running an architecture review' (Kazman & Bass, 2002). Issues of a social, psychological and managerial nature all impact upon the analysis process where it is inclusive of a broader stakeholder group (Kazman & Bass, 2002).

The architecture-based analysis literature suggests these issues can be resolved by properly setting the expectations of the participating parties, ensuring documentation is made available and that the facilitator is sufficiently skilled (Kazman & Bass, 2002). The focus on the facilitation, rather than the method itself affirms Jackson's view that traditional engineering design is focused on the 'systems' dimension of complexity at the expense of the 'people' dimension (Jackson, 1988). The 'people' dimension of complexity encompasses notions of psychological complexity (e.g. capabilities, notions, and perceptions) and metaphysical complexity (e.g. values, beliefs) (Flood, 1987).

Methods like the Architecture Tradeoff Analysis Method (ATAM) inherently explore competing goals but don't appear to adequately address the underlying beliefs influencing the process. Work by Sargent suggests that the negotiation of goals needs to partnered with the negotiation of meaning (Sargent, 1994). In the context of architecture-based analysis, the focus is clearly on the identification of competing properties within the architecture. The negotiation of meaning is not considered important to the process even though its role as improving communication between the stakeholders is acknowledged. The processes, therefore, appear deficient in specifying how they rationalise all of the viewpoints into what Flood describes as a unitary output (Flood, 1988). Indeed the strength of facilitation and consulting like manner in which the parties interact has the danger of creating artefacts in the likeness of the evaluation team, which neither the design team nor client teams fully understand.

Given the insufficient understanding of stakeholder complexity on architecture analysis the following thesis seeks to examine the dimensions of complexity in architecture-based analysis and how they impact upon the process itself. A particular emphasis will be placed on the different ways in which participants view the system at hand, acknowledging the prominent design theoretic views on personal perspectives in design tasks (Bucciarelli, 2002). The research utilises the collaborative project environment to provide an in-depth look at qualitative aspects of the process, which is seldom addressed within the literature.

1.2.Methodology

The research environment of the collaborative project and the specific research focus required a method supportive of two roles: the first, a researcher role in the next-generation network (NGN) management system design project; the second, a role specifically researching architecture-based analysis. These two roles did not perfectly overlap as the first required a broader scope aimed at developing the NGN management system architecture. Action Research (AR) as a methodology for situated inquiry is sympathetic of the need to perform both roles and is also accepting of change as a mechanism of developing further understanding. As addressed in the section 1.3 the need to be adaptive is vital to the research project as the architecture-based analysis was designed firstly as a mechanism of furthering the design of the system and secondly as the primary subject of research for the researcher.

The need for both action and learning is revealed in the structure of the methodology. In its most abstract form, action research consists of: stages of planning action; taking action; and reflecting upon action. These phases form a natural cycle in which the reflection and learning from the previous cycle influences planning and action in the next phase.

The structure of the methodology is reflected in the composition of the chapters concerned with the action research cycles themselves. Evidence of the affect of the methodology on the thesis content is also apparent in other aspects of the writing from the explicit discussion about learning on the methodology in chapter 7, to the comparatively 'fuzzy'[2] research objectives in the following section. Significant detail of the methodology is deferred to the appropriate chapter on the 'research approach'.

1.3.Research objectives

It is intended that the research begin with no clear hypotheses, but simply a situation in which understanding and change are both desired in reaction to a perceived problem. In this instance, the situation consisted of the need to make use of architecture-based principles to address the quality requirements of an NGN management system. The research objectives reflect the focus of understanding which is designed to address a perceived problem and facilitate effective change. As discussed above the problem was expressed as schism between the focus of architecture analysis methods and the reality of the design situation. The reality of the design situation was highly diverse and influenced by individual stakeholder perspectives.

---

[2] http://www.scu.edu.au/schools/gcm/ar/art/arthesis.html (Dick 1993)

In accordance with the goal of assuring the management solution would fulfil its quality expectations. As discussed in the motivation section above, the specific focus on architecture-based analysis arose from a combination of practical exposure to the design situation and the impending need to apply the architecture-based analysis (A-bA) techniques within the design process. A perceived schism between the concerns and capability of the A-bA methods and the characteristics of the design situation sparked apprehension as to their efficacy, prompting a need for greater understanding.

Consequently this research aims to:

"Understand and manage the dimensions of complexity in architecture-based analysis using the action research methodology."

The research is expected to generate learning in the following key areas:

- *The manifestations of complexity in the architecture-based analysis process and what are their effects on the objectives of the process?*
- *The effective methods of managing the dimensions of complexity in architecture-based analysis?*
- *The iterative application of architecture-based analysis throughout the design lifecycle.*
- *The application of action research to the field of architecture-based analysis.*

1.4. Thesis structure

1.4.1. Diagrammatic representation

Figure 1 below illustrates the structure of the thesis and indicates some salient relationships between the chapters.

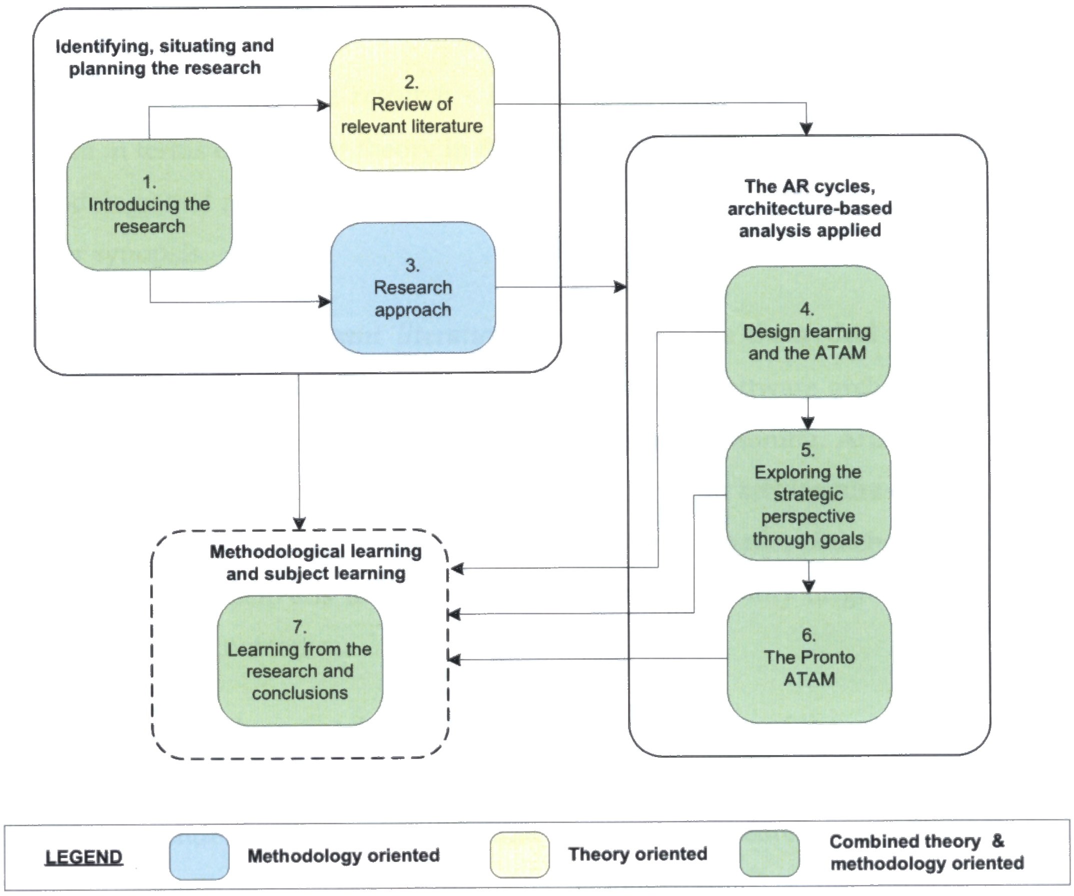

Figure 1.1 - Thesis Structure

As shown in the top left hand side of Figure 1.1, the first 3 chapters of this thesis identify, situate and the plan the research. Chapter 1 and Chapter 2 identify the research situation and situate it within both a practical and theoretical context (identifying and situating). Chapter 3 discusses the way in which the research will be conducted with respect to a chosen methodology (planning). The theory and methodology then form the basis of the action research (AR) cycles of chapters 4, 5 and 6 (right hand side of Figure 1.1). The cyclical nature of the methodology is evident in this part of the figure with each action cycle informing the next. From these action research cycles, the most salient outcomes are discussed. Learning is presented on both the application of the methodology as well as the subject area (bottom left hand side of Figure 1.1).

1.4.2. Summary of chapter contents

Chapter 1: Introducing the research - discusses the motivation for undertaking the research in terms of relevant theory in the area and the research situation at hand. The methodology and research objectives are outlined as well as a document structure and chapter synopsis.

Chapter 2: Review of relevant literature - establishes the theoretical context of the research through the presentation of relevant literature. Software architecture (SA) is introduced as a discipline to enable early design stage reasoning. Architecture-based analysis is presented in detail. An examination of trends in architecture-based analysis establishes a trend in analysis towards incorporating a broader stakeholder group. A soft systems perspective is then used to understand the ability of group diversity to affect the analysis outcomes.

Chapter 3: Research approach - discusses the characteristics of the research situation and the research objectives, concluding that a method of situated inquiry is critical to developing understanding commensurate with the research objectives. Several available research approaches are presented amongst which action research (AR) is argued to be the most appropriate. The rest of the chapter outlines the specifics of AR and how it can be best applied to the research situation at hand.

Chapter 4: Design learning and the Architecture Tradeoff Analysis Method (ATAM) - recollects the first action research cycle where architecture-based analysis is used by the project team to examine existing solution approaches. As this is the first AR cycle, Chapter 4 presents the research situation in more detail and constructs an initial framework of knowledge in accordance with the need for methodological rigour discussed in Chapter 3. The Architecture Tradeoff Analysis Method (ATAM) is selected as the most appropriate method of analysis for the project and the benefits of its use early in the design cycle established. The remainder of the Chapter reflects on the experience and utilises the framework of knowledge to structure learning.

Chapter 5: Goal-based requirements (GBR) and the strategic perspective - explores the strategic perspective in a bid to incorporate it formally into the requirements so it can be used in future design and analysis activity. The concept of goal-based requirements is discussed and an appropriate method is chosen to conduct the exercise.

Chapter 6: Project architecture analysis - presents the final AR cycle for the research project. This final cycle utilises architecture-based analysis to analyse the project team's solution architecture codenamed 'Pronto'. Particular interest is taken in the effectiveness of using scenarios in the ATAM to build understanding during the conceptual stages of design. The significant amount of time elapsed since the earlier analysis cycle also provides insight into the use of the ATAM at different stages of systems design.

Chapter 7: Learning from the research and conclusion - summarises the work and addresses the research objectives outlined in the Chapter 1. Chapter 7 also discusses future research opportunities which stem from the work presented in this thesis.

1.5. Original contributions

The author believes that the following areas of research outlined in this thesis represent contributions to knowledge:

1. The first contribution is made with respect to architecture-based analysis in Chapter 2.

    a. <u>Critical analysis of existing architecture-based analysis techniques against a framework for complexity.</u> Section 2.3 and Section 2.4 presented the complexity of evolved methods of analysis with respect to both systems and people complexity.

    b. <u>Establishing a soft systems perspective on architecture-based analysis.</u> Sections 2.5 and 2.6 establish the need to manage complexity associated with people within the process itself, rather than as an external facilitation exercise.

2. The second contribution is made in Chapter 4 and Chapter 6 with respect to the research methodology and the Architecture Tradeoff Analysis Method (ATAM).

a. The use of Action Research (AR) to investigate the application of the ATAM. Action Research is seldom applied within the field of software architecture and no literature precedent was found for specifically researching the ATAM using AR.

3. Techniques to manage complexity in the ATAM. The desire of this research was to understand the complexity before promoting techniques to resolve it, however the need to overcome difficulties resulted in the presentation of some methods to assist in the analysis, these include:

   a. The application of Goal-Based Requirements (GBR) to understanding the strategic perspective during architecture-based analysis. Goals are used to structure the business drivers, which form the inputs for the derivation of system quality requirements in the ATAM (Sections 5.5 and 5.6).

   b. The use of a Goal-Quality matrix to assist in constructing the utility tree. A matrix method of relating goals and system qualities to help scenario generation is discussed (Section 6.3).

4. The most significant contributions are made in the final chapter, Chapter 7. The value of understanding in situated inquiry is prominent in the learning outcomes of Chapter 7, which contributes to the understanding of:

   a. The complexity of conducting the ATAM. The aspects of complexity in applying the ATAM are clearly documented against the theoretical framework (Section 7.1.1).

   b. Consequences of situational complexity for the ATAM process. The affect of this complexity on the objectives of the ATAM is presented (Section 7.1.2).

   c. ATAM and the design lifecycle. Section 7.1.3 presents findings on the use of the ATAM at different stages in the design lifecycle. Based on the understanding from this project, a method of analysis addressing two diverse, but related perspectives is proposed.

1.6.Refereed Conferences

The following refereed papers, relating to the thesis material, have been published and where appropriate presented by the author. Of these, selected key papers are reprinted in full in Appendix A.

Colquitt, D. and Leaney, J. *'Expanding the view on complexity within the Architecture Tradeoff Analysis Method'* 14th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS), Tucson Arizona, March, 2007.

Colquitt, D., Leaney, J. and O'Neill, T. *'The case for understanding social complexity in the architecture-based analysis process'* 1st International Conference on Qualitative Research in IT (QualIT), Brisbane Australia, November, 2004.

Colquitt, D., Leaney, J. and O'Neill, T. *'Integrating Architecture-based Trade-off Analysis into the design process through tool-assisted modelling'* 2nd IEEE International Workshop on Model-Based Development of Computer- Based Systems (MBD), Brno Czech Republic, May, 2004.

Sheridan-Smith, N. B., Colquitt, D., Soliman, J., Leaney, J. R., O'Neill, T. and Hunter, M. *' Improving The User Experience Through Adaptive and Dynamic Service Management '* Australian Telecommunication Networks and Application Conference (ATNAC), Sydney, Australia, December, 2004.

Summary

Chapter 1 provided a broad introduction to the thesis outlining the motivation for the research, the resulting research objectives and the overall structure of the thesis. The motivation for this thesis is presented as arising from the early experiences within a collaborative project to design a next generation network (NGN) management system. The strong diversity of the design group was discussed as challenging the existing notions of complexity within architecture-based analysis methods. With the research group needing to utilise architecture analysis, the research is shown to contribute to a practical project need and a deficiency in current architecture-based analysis theory. The methodology was then briefly introduced because of its influence on the structure of the thesis and the research objectives. Following this the research objectives were specified and the structure of the thesis established. Having established the research situation and objectives in Chapter 1, a thorough review of relevant literature will be presented in Chapter 2.

2

Review of relevant literature

Introduction

In Chapter 1 the research theme was presented and some background motivation given for undertaking the research. The motivation discussed the exacting quality requirements of future telecommunications management systems, and the complementary quality focus of software architecture. Architecture-based analysis was established as the focal point of the research following early experiences of group diversity in design meetings. Chapter 1 concluded that existing architecture-based analysis techniques did not manage the diversity of stakeholders and that further learning and research was required in the area.

Chapter 2 reaffirms the need for a discipline like software architecture to enable early stage design reasoning. Architecture-based analysis is presented in detail, outlining its original role in selecting candidate designs and subsequent evolution to incorporate a broader design learning role. The subsequent expansion of the stakeholder group accompanying this evolved purpose is discussed as increasing the dimensions of complexity associated with the process. A soft systems perspective is then used to shed light on the uncertainty of the analysis process and how group diversity can significantly influence the analysis outcomes. Finally the incorporation of the soft perspective into the analysis task is presented as promoting further investigation as to how the process is impacted by and subsequently copes with complexity that extends beyond the traditional 'systems' view.

The structure of Chapter 2 is shown in Figure 2.1.

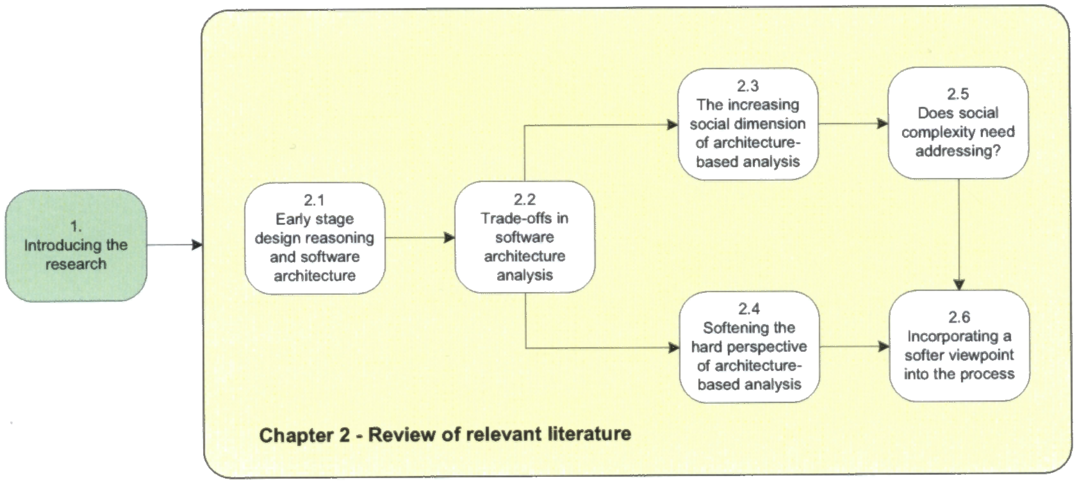

Figure 2.1 - Structure of Chapter 2

2.1. Early stage design reasoning and software architecture

A natural part of the design process is the making of decisions both consciously through deliberate design choices and implicitly through commitment to the decision and acceptance of its consequences (Schön, 1991). The consequences alluded to by Schön have many dimensions, two of which are the new issues raised in realising the decision and the pruning of previous options from the 'tree of moves' (Schön, 1991, pp 100), referred to as a hierarchy of interdependent sub-solutions (Joseph, 1996). In the context of software design, the notion of deepening commitment is strongly supported by some fairly cogent arguments about the cost of changes to early design decisions. Boehm has estimated the cost of late correction of requirements errors within a partially or fully developed system increases 200 fold when compared with the cost of early correction within the requirements engineering process (Boehm, 1981). Likewise it has been shown that changes to the fundamental structure of an implemented system are the most costly form of evolution (Brooks, 1995; Rechtin, 1991).

A logical inference from these observations is the further along the system design and development path, the more committed one is to the solution, and the more costly it becomes to change earlier design decisions (Abowd et al., 1997). The flow on effect of design decisions, coupled with the cost of late changes, raises the profile of the earliest design decisions as being critical to the efficacy of the end solution. The situation is analogous to the built environment where the underlying structure (architecture) contributes most significantly to the properties of the end product (construction). The relationship between the two fields has led to the popularisation of the term software architecture (Land, 2002; Perry & Wolf, 1992).

Software architecture is described as the *"the principled study of the overall structure of software systems, especially the relations among subsystems and components"* (Shaw, 2001, pp 657). Although the notion of an 'overall structure' has been challenged as encouraging the narrow view that one structure defines a system (Northrop in Bass et al., 2003), this definition highlights two critical aspects of software architecture. These two critical aspects are those of 'structure' and 'principles'. In terms of software architecture, the structure is an abstract representation of the system. The abstraction is designed to suppress details of any component that do not affect how they relate to, or interact to other elements (Bass et al., 2003). By encouraging representations of the design at an abstract level, software architecture facilitates early communication amongst stakeholders. Importantly the abstract design provides a basis for reasoning about system quality concerns from the earliest system form onwards (Bass et al., 2003).

Abstract design representations are an important aspect of software architecture, nevertheless designers had been producing designs at many levels of abstraction well before software architecture was envisaged. It is the set of principles derived from the disciplined observation of real world design that differentiates software architecture as a practice distinct from the 'box and line' diagrams of preceding design behaviour. Having at its core the representation of systems, it is no surprise that many of the prominent principles in software architecture concern aspects of building and evolving models. These principles range from guidance on describing architectures (IEEE, 2000), to what types of views and entities could and should be represented (Kruchten, 1995), to formal architecture description languages (ADL)s for expressing and manipulating architectures in an unambiguous fashion (Medvidovic & Taylor, 2000).

While all these developments are important to the practice of software architecture, the cornerstone of the discipline are the earliest principles. These principles developed through the observation that component types and patterns of interaction tended to reoccur across disparate designs. Termed 'stylistic commonalities' these design patterns clued practitioners into the idea that specific structures could be repeatedly applied to solve a particular class of problem (Shaw, 2001). The further generalisation of this concept across different problem domains led to the development of architectural styles, which are effectively configurations of component types and their patterns of interaction (Bass et al., 1998).

The importance of these styles to the broader software architecture and design communities was the fact that styles captured strategies of fulfilling functional and quality objectives alike. As discussed in Chapter 1, traditional systems design focussed on the functional goals of systems, leaving a need to understand quality expectations of clients and how they can be satisfied through design decisions. Architectural styles posited a highly significant relationship between the goals of a system, be they functional or quality related and the earliest (abstract) system structure. Importantly it promotes the ability to design for performance, maintainability, security, etc through architecture-based decisions.

2.2. *Trade-offs in software architecture analysis*

Inherent in the need to design for specific goals of a software system is the need to test for them. Software architecture principles help create a degree of determinism between design moves and system properties, but it is by no means a 'cookbook science' (Bass et al., 2003), immune to the complexities that arise in system design situations. Styles may have made a significant contribution to the way in which the architecture-based design of software systems can be approached, but their application lies within fulfilling a specific set of requirements which are inherently a subset of the overall system needs. Competing system quality requirements are inherent to systems design engendering trade-offs that need to be analysed and resolved with respect to the client's objectives (Bass et al., 1998; Kazman et al., 1999). Reference to client objectives is critical as there are no guarantees that the criteria for engineering success are the same as those for operational success. A design that meets all of its specifications and constraints can still be perceived as a failure by its users (Kroes, 2002). Consequently, the effect of design decisions on the capability of the end system needs to be iteratively (and incrementally) tested against the expectations of the key stakeholder groups (Bachmann et al., 2000).

Without the ability to evaluate architecture-based design decisions against the quality attributes, an architectural approach offers little benefit over existing methods of software engineering. Shortcomings in the design will remain undiscovered until the later stages of implementation, incurring the same costly penalties alluded to earlier in this chapter. In order to satisfy the need for assessing software architectures many different techniques have been developed. Amongst these methods Abowd has identified two significantly different types of architecture-based analysis. The dichotomy is split along the lines of questioning and measuring techniques (Abowd et al., 1997). As indicated by the name, measuring techniques use specific measures and models of system behaviour (be they prototypes, simulations, etc) to yield quantitative results. Alternatively questioning techniques utilise less specific methods such as scenarios to develop a qualitative understanding of the system and how it fulfils its requirements (Kazman et al., 1999).

Although there are no hard and fast rules about which type of analysis is preferable given a specific design situation, there are several facets of the measuring techniques that tend to dissuade their application in complex design situations (Abowd et al., 1997). The first is referred to as a lack of generality, in that measuring techniques are designed to answer specific questions about an architecture and are only applicable to specific qualities at any one time, precluding the ability to analyse trade-offs (Abowd et al., 1997; Kazman et al., 1999). Secondly the level of detail required to develop faithful behavioural models is often infeasible due to the amount of resources required to produce such models and the limited system understanding early on in the design lifecycle. In the event the model is an actual prototype considerable design work would already have been conducted encumbering the ability to roll-back and explore other design alternatives on the basis of the results. Properties like these do little to facilitate the notions of iterative and incremental design that are commonly applied to complex design problems.

The perceived inflexibility of the measuring techniques has since been reflected in the favoured development of questioning methods of architecture-based analysis.

Although this is not to say that the nature of questioning techniques is inherently compatible with design needs of software architecture. Indeed many of the early qualitative techniques bear the hallmarks of their quantitative counterparts, aspiring to ambitions of objectivity and repeatability (Hilliard et al., 1996). In the earliest methods these ambitions were satisfied by incorporating the assignment and use of numeric values and weightings. Methods like Rank Matrix Analysis (RMA) (Hitchins, 1992) and Quantified Design Space (QDS) (Shaw & Garlan, 1996) utilised matrices to facilitate the numeric evaluation, whilst methods like the Software Architecture Analysis Method (SAAM) (Kazman et al., 1994) and Architecture-based Quality Assessment (AQA) (Hilliard et al., 1996) manipulated scores to produce overall measures of effectiveness.

These methods were very much selection oriented in that they took requirements and design configurations as unquestionable statements of system purpose and structure, and then sought to score and select specific design approaches that best suited the requirements. Difficulties in reliably scoring system designs aside (Hitchins, 1992), these analysis techniques did not foster understanding about the root causes of perceived design problems with respect to the customer objectives. Ultimately they provided a means of selecting between a set of candidate solutions, but allowed no further learning as to how the end solution could be improved to account for inconsistencies encountered during analysis (Lee & Hsu, 2002). Effective as selection tools, these techniques were ineffective as design project tools. Ineffective because they are unable to facilitate incremental analysis throughout the design lifecycle and provide input into future design iterations (Houkes, 2002).

Consequently more recent methods of architecture-based analysis, including the Architecture Trade-off Analysis Method (ATAM) (Kazman et al., 2000) and the Software Architecture Review Assessment (SARA) (Obbink et al., 2002), have sought to distance themself from checklists and the numeric assignment of values by declaring a focus on architectural risk. The reality of designing according to the ATAM is that the perfect system is unattainable due to requirement conflicts. Design should provide a way of trading off these requirement conflicts in a way that still achieves the client's goals. Instead of simply selecting amongst candidate design options ATAM promoted the development of customer goals; the association of these goals to the system quality drivers; the documentation of design strategies to fulfil these drivers; and the identification of points in the architecture where multiple quality attribute concerns intersected. By identifying aspects of the design requiring greater care and fostering further understanding of both the requirements and design approach, methods like ATAM have evolved to fulfil not just an evaluation, but an effective analysis role.

2.3.The increasing social dimension of architecture-based analysis

The ability of ATAM to provide an inexpensive (Kazman et al., 1999) method of analysis that can be used throughout the design process (Dobrica & Niemela, 2002) has realised a much tighter integration of architecture-based analysis into the overall design process (Bachmann et al., 2000). Therefore it is not surprising to see architecture-based analysis following the pattern of broader design methods building a platform for the democratisation of the design process (Joseph, 1996). The progression away from techniques of candidate selection towards fulfilling a design analysis role has moved the process from the domain of a few technical experts, to that of the broader stakeholder community. A community which has grown proportionately to the number of stakeholders whose opinions legitimately shape the form of the design. This diverse group is described esoterically by Bucciarelli as a 'design collective' (Bucciarelli, 2002).

Similarly the role of architects is continually being revised and expanded in light of their need to balance the individual interests of the ever expanding design collective. The consideration of system qualities includes the more traditional design considerations such as performance and availability, but also opens the door on any number of imaginable attributes. Properties such as cost, time, usability, and safety, can all be naturally reasoned about with relation to the structure of the system.

> *"When Brunel and Robert Stephenson were building railways in the late 1830s and 1840s, they were expected to involve themselves with raising capital, appearing before Parliamentary Committee, conciliating influential people..........Why should we be surprised if Software Engineers may need to draw on expertise in mathematics, financial analysis, business, production, quality control, sociology and law, as well as in each application area they deal with"* (Jackson, 1995)

Jackson's software engineer as bricoleur[3] is highly telling of the need to balance more than purely technical issues in engineering an effective system. Similarly when trying to evaluate what is an effective system there needs to be adequate consideration of such concerns. ATAM and SARA, widely viewed as the industry best practice methods (Obbink et al., 2002) both strive to give representation to all key stakeholder groups. These methods acknowledge the contribution of stakeholders to realising an effective design and the importance of achieving greater levels of understanding and communication within the group (Bass et al., 2003). In doing so they also bring upon themselves concerns associated with managing *"the non-technical aspects of running an architecture review"* (Kazman & Bass, 2002, pp 67). Issues of a social, psychological and managerial nature all impact upon the analysis process where it is inclusive of a broader stakeholder group (Kazman & Bass, 2002).

The extent to which these concerns are understood and handled in the context of architecture-based analysis are conspicuous by their absence with only recent acknowledgement, *"as architecture reviewers, we continually run into social, psychological, and managerial issues and must be prepared to deal with them."* (Kazman & Bass, 2002, pp 68). Kazman at al suggest resolution to these issues should occur through successful facilitation and process management, echoing several points from their literature about needing to negotiate entry into an organisation and effectively set expectations (Clements et al., 2002). Several pragmatic facilitation skills are also put forward as being integral for conducting a successful analysis. Amongst these are the needs to *"control the crowd, involve the key stakeholders, engage all participants, maintain authority, control the pace, and get concurrence and feedback"* (Kazman & Bass, 2002, pp 71).

While these behavioural aspects of group dynamics are important to the effective functioning of the group, they don't touch on how the method itself might be adapted to mitigate these issues. Social, psychological and managerial issues are human aspects to the process and might well be seen to only affect relationships within the group, leaving aside potential affects upon analysis artefacts. However as the following

[3] Bricoleur is a French word effectively meaning "jack of all trades" or "handyman"

sections attest there are significant areas of related research that entertain a stronger consequence of human issues on social processes.

2.4. Softening the hard perspective of architecture-based analysis

Design theoretical and methodological research offers another dimension to the characteristics of social processes, presenting the view that *"we see reality through the mental filter of our 'ideas' or conceptions. If we accept this commonplace observation it is hard to see how one could ever talk about reality except through the very same filter."* (Galle, 1999). Here Galle touches on a significant topic associated with human perspective and understanding, which has a well respected lineage in the form of 'Weltanshauungen'[4] (Checkland & Holwell, 1998; Hitchins, 1992), 'holons'[5] (Checkland & Holwell, 1998), 'psychological and metaphysical complexity' (Flood, 1988) and 'object worlds' (Bucciarelli, 1994).

In organisational development terms, the social system created by the collection of individuals needs to be considered as a soft system. Sir Geoffrey Vickers fostered the softening of hard systems thinking towards group dynamics (Checkland & Holwell, 1998). The previous view of organisations was that the group had a common goal and understanding and were working to achieve that goal through decision making. Soft systems thinking introduced the notions raised above about individual motivations, experience and views of the situation that needed to be both understood in context of their peer's world views and accommodated for in decisions (Checkland & Holwell, 1998). Rather than relegating such concerns purely to the domain of human affairs, the field of information systems (IS) has shown that the 'messiness' (Hitchins, 1992) associated with such systems extends to domain of purposeful human activity systems

[4] Used commonly in the work of Checkland to mean world view, Weltanschauung is a calque of a German word meaning "a look onto the world". It refers to the framework through which an individual interprets the world and interacts in it

[5] Coined by Arthur Koestler (The Ghost in the Machine 167), Holons are used to describe something that is simultaneously a whole and a part of a larger system, for example a letter is a entity but also part of a word. Checkland uses the concept to describe the relationship between the models we create of the world around us and the world itself.

(HAS) incorporating both the use (Hitchins, 1992) and design contexts (Jagodzinski et al., 2000) of information technology. Extending this notion, it does not seem problematic to suggest that evolved methods of architecture analysis cannot escape the characteristics which would see it viewed as a social process akin to a 'messy' human activity system (HAS) (Hitchins, 1992). The elements of hierarchy, different domains of concern (Obbink et al., 2002), different historical perspectives and experience, different intentions (Galle, 1999), different perceptions of the situation (Janes, 1988), social disharmony are all relevant to the architecture analysis process as much as they are the design process at large.

These elements tend to proliferate where the system at hand is strongly defined by the people within it, as such the continual interaction and change in the environment makes the system form somewhat intangible. The argument could be made that the unambiguous ways in which technical systems can be described could potentially resist or at least diffuse extended notions of complexity. However the certainty with which we treat extant artefacts is not necessarily reflected in the nature of their representations or the processes that yield them. When dealing with technology the temptation is to treat the process in the same light as the product. In Boulding's classification of systems, structures are classified as physical or mechanical systems, 'i.e. hard, and are in the province of the physical sciences' (Hitchins, 1992). However the journey from concept (design need) to design artefact (communicative medium) (Bucciarelli, 2002) to system or structure does not resemble the characteristics of the end product at all. In terms of design, all that exists are representations of concepts of the system which are in turn interpreted by the stakeholders (Galle, 1999). The use of design representations as a means of communication (Galle, 1999), places the process at the 'social' end of Boulding's classification.

Specifying purely facilitator behavioural traits as the mechanism for managing social complexity within a process is noticeably dismissive of any need to adapt the process itself. The objectivity (Hilliard et al., 1996) and replicability (Kazman et al., 1994) that were the ideals of earlier analysis methods appear not to have changed. The same theoretical perspective that informed earlier beliefs about architecture-based analysis is still thought to hold even in the face of 'psychological complexity' and the theoretical arguments about the interpretive nature of design (Galle, 1999). Reasoning for such a perspective can be found in the emergence of software architecture amidst the disciplines of software and systems engineering, which are the domain of traditional 'hard' systems thinking processes (Jackson, 1988).

Hard systems thinking seeks to apply a scientific approach to the complexity of real world problems, working from the assumption that *"the problem task they tackle is to select an efficient means of achieving a known and defined end"*(Checkland 1978 in Jackson, 1988, pp 155). The selection of means takes place through the development of quantitative models of the situation, incorporating all factors of relevance, which then form the basis of 'experiments' to determine an optimal solution (Jackson, 1988). Approaches like this initially suited disciplines like software engineering because the availability of a prototype, or potentially the system itself, allowed clear results to be obtained (Bosch & Molin, 1999). However the discipline of software architecture seeks to intervene earlier in the design process to avoid effort being poured into a system that cannot fulfil its quality requirements. The abstract nature of software architecture complicates the 'selection of means' as it is not possible to measure the qualities of the end system based on the architectural design. Instead the focus is on analysing its 'potential' to reach the required level of quality.

Likewise the assumption of a 'defined end' in designing computer-based systems has traditionally proven to be problematic. Architecture-oriented design processes tend to depict architecture-analysis as being informed by a comprehensive requirements engineering exercise (Figure 2.2 and Figure 2.3). However the ability of requirements exercises to provide a firm platform for analysis needs to be questioned in light of uncertainty often encountered during requirements engineering (Antill, 1986).

> *"...it is really impossible for a client, even working with a software engineer, to specify completely, precisely, and correctly the exact requirements of a modern software product before trying some versions of the product"* (Brooks, 1987)

Adding to the requirements problem is the fact that quality attributes are a more recent concern in systems design and are commonly represented and reasoned about in a vague manner.

"In a perfect world, the quality requirements for a system would be completely and unambiguously specified in a requirements document .... In reality, requirements documents are not written, or are written poorly, or do not properly address quality attributes." (Kazman et al., 2000)

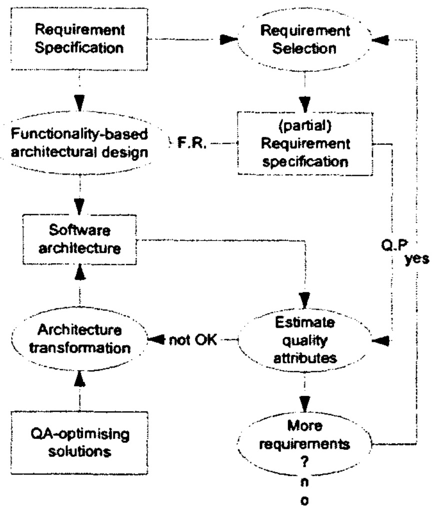

Figure 2.2 - Quality attribute oriented software architecture design (Bosch, 2003) process

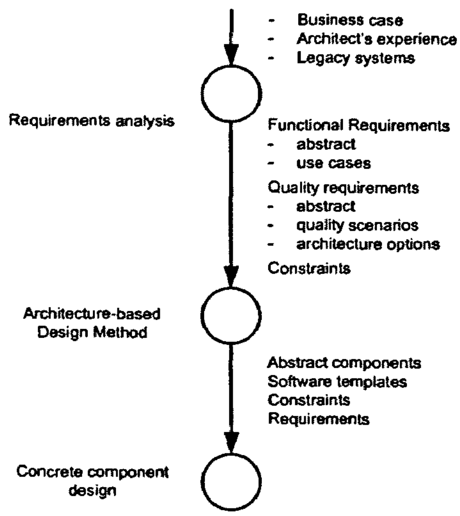

Figure 2.3 - ABDM (Bachmann et al., 2000)

In disciplines where there are 'unambiguous ends' and 'fixed contents of professional knowledge sufficient for rigorous practice', there is perhaps a diffused impact of social complexity (Schön, 1991). The clarity of the ends and means provides for a well formed problem that may be solved by a 'calculus of decision' (Schön, 1991). Although there have been some attempts at relating structural measures to quality attributes (van Gurp, 2000), accompanied by the declaration of several design heuristics such as Attribute-Based Architecture Styles (ABAS)s (Klein & Kazman, 1999), it can be said that few irrefutable or un-situated truths currently exist in the world of architecture-based analysis. Rather than presenting as well formed, its problems must be constructed from difficult situations. The uncertainty surrounding the definition of 'ends' (requirements) and selection of 'means' (design approach), suggest there is scope to incorporate a softer view of the architecture analysis process that acknowledges the true complexity of the task.

2.5.Does social complexity need addressing?

As is the case with Schön's architects of the built environment and their sketches, systems architecture deals in the realm of virtual worlds.

> *"The situations of Quist and the Supervisor are, in important ways, not the real thing. Quist is not moving dirt on the site. The Supervisor is not talking to the patient. Each is operating in a virtual world, a constructed representation of the real world of practice"* (Schön, 1991)

Similarly in architecture-based analysis, the architecture presented to the stakeholders is a partial representation of the system, from which they are left with the task of mentally constructing the system, its goals and importantly their intent for it. These world views both unite stakeholders in some aspects and divide them in others, for the view that they share the same object worlds has already been rejected (Bucciarelli, 2002). Davies suggests that the metaphysical complexity introduced in situations such as examining complex virtual systems is dealt with in human terms by 'human sense-making', simplifying the world by selecting from it *"that which it takes to be important aspects of that world"* (Davies, 1988).

> *"This is the selection of relevance from the world via an assimilation and accommodation process"* (Piaget 1952 in Davies, 1988, pp 134).

Soft systems methodology maintains that this accommodation needs to be reached in a group sense, through a common understanding of the system at hand, and an appreciative understanding of the individual world views (Weltanschauungen) of the stakeholder group (Checkland & Holwell, 1998). Only when accommodations are made and a sort of group understanding formed, can the target system be reasoned about. Without this common understanding, the individual contributions can conceptually swamp the process, imposing their view upon the situation and adding to the situational complexity rather than seeking to resolve it (Davies, 1988).

Bass suggests that the stakeholders have a 'limited' role in 'crafting' the goals for the architecture and subsequent scenarios (Bass et al., 1998). This reduced participation appears to mitigate further problems experienced by involving the stakeholder community. However the way in which the stakeholders view the system, their intended uses for it, and their overall goals for the system are the critical benchmarks that drive the analysis methods. Understanding these factors with respect to the stakeholder group is imperative to the success of the analysis process, something which appears to be jeopardised by the existing lack of consideration for managing social complexity within the architecture-based analysis process. Being the medium through which the stakeholders communicate, architectural representation is a logical nexus of viewpoints and concerns for the design process. Architecture analysis acts as a key integrating component serving to both further explore the problem space by expounding the undeclared goals of the customer, as well as provide guidance for the architects in attempting to realise a satisfactory solution.

Recalling the earlier discussion of Jackson's software engineer as bricoleur and the social behaviour guidelines for the ATAM, it becomes evident that systems architecture has placed the responsibility for managing social complexity on the crowded shoulders of the facilitator. The focus on the facilitation, rather than the method itself, as addressing all of the non-technical issues affirms Jackson's view that traditional engineering design is focused on 'systems' complexity, at the expense of 'people' complexity (Jackson, 1988). The people dimension of complexity encompasses notions of psychological (capabilities, notions, perceptions) and metaphysical (values beliefs) complexity (Flood, 1988). In many ways augmenting the importance of facilitation can be counter-productive to the process of building understanding. The need to beware of an apparent 'mismatch' in the communication chain of architecture-based analysis has been remarked in ATAM literature.

> *"...even though the review team is frequently the focus of the conversation and the source of many of the probing questions. The review's outputs are really for the stakeholders—the review team members are just there to act as catalysts, experts, and facilitators. ..."* (Kazman & Bass, 2002, pp 72)

Architecture-based analysis is essentially charged with juxtaposing the problem owner's position with that of the designer to ensure that they align. Design (Architecture) artefacts are the means through which they communicate and negotiate understanding of each other's object worlds. A negotiation that needs to take place within a social framework (Sargent, 1994). In demonstrating that negotiation plays a central role in design activities, Sargent revealed that negotiation of goals is only one of two significant dimensions, the second being the need to negotiate meaning (Sargent, 1994). In the context of architecture-based analysis, the focus is clearly on the identification of competing properties within the architecture, the resolution of which is deferred to another process (Bass et al., 2003). The negotiation of meaning is not considered important to the process even though its role as improving communication between the stakeholders is acknowledged (Obbink et al., 2002).

2.6. *Incorporating a softer viewpoint into the process*

Despite the legitimate case made for the existence of social complexity in architecture-based analysis, there is little indication from the literature or case reporting that consideration has been made for it within the published process. Instead consultation and facilitation are put forward as the mechanisms by which such issues are handled when conducting ATAMs. Apart from some pragmatic skills for the general facilitation of analysis workshops, there is little clarity of how grey areas in the process such as the initial consultation to develop artefacts to an adequate level of maturity, or oversight of artefact creation are achieved to the satisfaction of the concerned stakeholders. Hard systems thinking tends to encourage the viewpoint that once the models (in this case the analysis artefacts) have been created, there is significant truth within them that the results are unambiguous. However a softer viewpoint of the process has been offered which suggests the artefacts created for, and used during analysis, cannot be looked upon with such certainty and will always be prone to perspectives of the participating stakeholders. Consequently managing the ways in which they communicate and negotiate with one another should be a core concern of the process itself, rather than an ancillary concern of facilitation.

To date the case study reporting of the ATAM literature tends to reflect the primacy of the process as it is published, rather than particularly as it plays out in the workshops. Whereas experience in the project highlighted the diversity of participating stakeholders and the extent to which such diversity affected design meetings, supporting a softer perspective on the process. The difference between this perspective and the existing understanding of the ATAM promotes further investigation as to how the process is impacted by, and subsequently copes with complexity that extends beyond the traditional 'systems' view.

Summary

The start of Chapter two discussed the evolution of architecture-based analysis from an expert-centric evaluation focus, to a stakeholder-centric analysis focus. This evolution has improved its utility in complex problem situations where iterative incremental lifecycles require analysis capable of furthering design understanding. The consequence of expanding the stakeholder group was then expressed in terms of opening up a dimension of complexity referred to as the people dimension. A discussion of related information systems and design literature was used to encourage a 'softer' perspective on what might be viewed in engineering terms as a structured technical task. Accepting such a perspective, even the more mature analysis methods were shown to provide little guidance on how to deal with difficulties arising from stakeholder diversity. Chapter 2 concluded that the lack of guidance, combined with insight into group diversity discussed in Chapter 1, promoted further investigation of how the process is impacted by, and subsequently copes with complexity that extends beyond the traditional 'systems' view.

Having established the research objectives in Chapter 1, Chapter 2 discussed in detail the associated literature in relevant research areas as well as further substantiating the research need from a theoretical standpoint. Consequently Chapter 3 will seek to establish the methodology to be used in the research project. Chapter 1 has already briefly described the motivations for the chosen methodology and discussed how its characteristics have influenced the presentation of the thesis material. Chapter 3 will provide a detailed discussion of available methodologies with respect to the research situation and carefully derive a research approach appropriate to the research situation and knowledge aims.

3

Research Approach

Introduction

Chapter 1 introduced the research situation of a collaborative project to design a next-generation network (NGN) management system. The motivation for the specific focus on architecture-based analysis was discussed and some appropriate knowledge aims presented. Chapter 2 provided a firm theoretical foundation for the research by reviewing relevant literature. Significant evidence was presented for a gap in architecture-based analysis knowledge, which could be addressed by the research aims. This chapter is concerned with developing a research approach suited to both the research situation and knowledge aims. Research precedents in software architecture are discussed and a particular type of inquiry selected as beneficial to both the maturity of the discipline and the research situation. A review of available research approaches is used to demonstrate the importance of reconciling the particular research method against the knowledge aims. A research approach is then constructed adopting a particular method and theoretical perspective. Chapter 3 concludes with a description of the research method and how it will be applied.

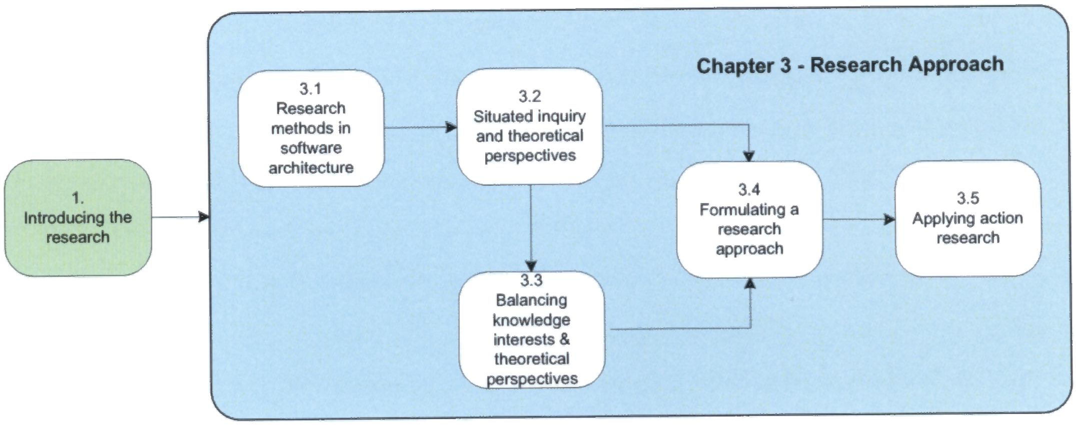

Figure 3.1 - Structure of Chapter 3

3.1.Research Methods in Software Architecture

Arising within the applied practice of software design it is not surprising to find that the approaches to research in software architecture are 'for the most part, the paradigms of software engineering' (Shaw, 2001). That is, research in both the disciplines of software architecture and software engineering exhibit similar characteristics. This section examines the characteristics of research in software architecture with respect to the broader field of software engineering, as well as the nature of the research as proposed in chapters 1 and 2.

In performing an examination on research in the field of software engineering Glass noted that *"Although the broader literature on research approaches is comprehensive, little has been published on appropriate ways of doing computing research"* (Glass, 1995, pp 3). Drawing upon observations of practice and an examination of relevant literature, he proposed a set of research phases that could be used to explain the patterns of research in software engineering (Glass, 1994). The set comprised of the informational phase (where information is gathered and aggregated), the propositional phase (where hypotheses are derived), the analytical phase (where the proposition is examined, demonstrating and/or developing a theory or principle) and finally the evaluative phase (where experimentation (controlled) or observation (uncontrolled) testing of the proposition take place).

These phases echo strongly through Shaw's examination of the research-based development of software architecture. In exploring the development of software architecture Shaw presented the progress of the field with respect to a model of the technology maturation process (Shaw, 2001). The model defines the phases of basic research, concept formulation, development and extension, internal enhancement, external enhancement and popularisation (Redwine & Riddle, 1985). Although the maturation model is based on the development of a technology towards popular adoption and the research phases are purely research focused the congruency is clear. Figure 3.2 shows the activities of the first three stages of each model as being similar in nature. The final three stages of the maturation model are accounted for in the iteratively expanding notion of evaluation that is incorporated into the evaluation phase of the research model (Glass, 1994).

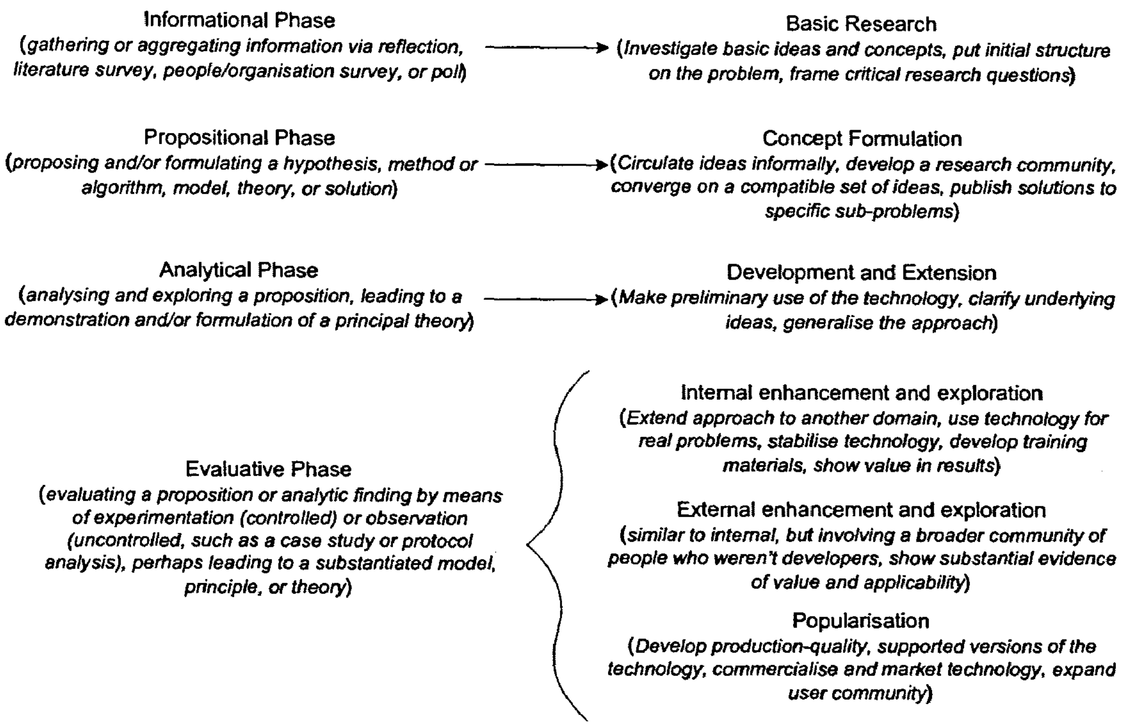

Figure 3.2 - Stages of the Glass research phase model juxtaposed with the Redwine technology maturity model as used by Shaw to describe software architecture

Following the progression of both these models, research begins with nascent ideas that are formulated and refined into rigorous concepts. These concepts are then tested in increasingly broader situations and eventually returned to the field of theory and/or practice. Faithful to the engineering method of research (Glass, 1995), software architecture develops its nascent ideas from the disciplined observation of practise and returns to the field what is essentially a better way of doing things.

From its roots in qualitative descriptions of useful system organizations, software architecture has matured to encompass broad explorations of notations, tools, and analysis techniques. Whereas initially the research area interpreted software practice, it now offers concrete guidance for complex software design and development (Shaw, 2001, pp 657).

In elucidating the importance of understanding practice within the field of software architecture before embarking on improving it, one comes to understand Brook's perspective that computer science is in fact 'not a science', but a *"synthetic, an engineering, discipline"* (Brooks, 1996, pp 62). The misnomer of computer science belies the fact it is concerned with 'making things' as distinct from the discovery of fact valued by scientists. A distinction exemplified by the activities of each discipline's exponents, whereby a *"scientist builds in order to study; the engineer studies in order to build"* (Brooks, 1996, pp 62). However the elegance with which engineering techniques can be reformulated in mathematical terms grants the impression there is a noble scientific pursuit disembodied from the real world (Sargent, 1994). The truth is that the techniques of design are not derivable by any detached/un-situated design science (Sargent, 1994). They are designed to solve real-world problems and will be judged by their usefulness and costs, not the novelty of their knowledge as is the case with traditional science (Brooks, 1996).

The challenge for software architecture research is that the complexities of the motivating problem are difficult to replicate in a research setting. In transferring the research theme from the real world of practice to a controlled research environment, simplifications have to be made that don't compromise the essence of the situation (Shaw, 2001). Removing detail from the problem situation, which is inherently achieved through the processes of characterisation and abstraction, permits the research outcomes a broader generality. As the characterisation and abstraction in the research laboratory begins, so does the estrangement of the technology from its origins within practice, as the theoreticians climb into their 'ivory towers' (Brooks, 1996).

In computer-based disciplines the dominant research approach has been demonstrated to be 'formulative' and the dominant research method to be 'conceptual analysis', with the only exception being the discipline of Information Systems (Glass et al., 2004). Research in this vein is focused on the development of theories or frameworks (formulative) (Morrison & George, 1995) and their proof through analytical methods (formal techniques) (Glass, 1995). Taking an analytic-formulative approach to researching architecture-based analysis techniques suggests the process can be examined, decomposed into constituent elements, and each of these elements reasoned about in respect to some objective ideal. In practical terms the process can be laid out on the table and examined, compared to notionally idyllic processes, the lacking elements changed and the process re-assembled. Such an instrumental view of engineering is seen as one of the key reasons why existing methods of inquiry are commonly viewed as inadequate and appear incapable of addressing the issues inherent in the 'people' dimension of complexity (Bucciarelli, 2002). Little concern is given to true empirical validation and case study observation. In reference to the set of research phases presented above the evaluation phase can be seen to be lacking, resulting in what Glass has termed rather unfavourably by his own admission as advocacy research (Glass, 1994).

Shaw offers a corollary to this in discussing the need for what she terms as 'validation' in software engineering. Expressing the view that research commonly exhibits inadequacy in two main ways. The first occurs when research aimed at improving practice ignores the need to collect evidence showing its application within practice. The second is when techniques are developed and tested by application to 'toy examples' with only tenuous links to a practical situation (Shaw, 2001). In order to prove validity research born technologies need to re-establish their merits within the complexity of the real world, *"good validation entails not only showing that the specific product of the research satisfies the idealised problem of the research setting, but also the result helps to solve the original motivating problem"* (Shaw, 2001, pp 660).

It is in the spirit of capturing the application of a technique to address the complexity of the real world that this research is largely undertaken. Architecture-based analysis has progressed through the phases of Glass' research model and is currently in the process of evaluation, embodied within technology maturity model as external application and enhancement (Shaw, 2001). The problem for architecture-based analysis is that many of the evaluation techniques suffer from the oversights in validation noted by Shaw. A search of the literature revealed few reports on actual experience in real-world application. Worked examples offered as evidence for working valid processes commonly deal with relatively simplistic examples. The only real method which has presented solid case material for its process is the ATAM. However the world view from which their experience to date has been conducted, appears insufficient in light of the discussion presented in chapters 1 and 2. By their own admission the main focus of reporting has been on technical aspects (Kazman & Bass, 2002).

The change in world view to incorporate the 'people' dimension of complexity promotes the idea there are difficulties the process has to overcome which are only attributable to the situation. That is the problems arise from the intersection between the technical and non-technical (Seaman, 1999), the interaction of the 'object worlds' (Bucciarelli, 2002) of the stakeholders with the task of architecture-based analysis. The human dimension of which urges the use of qualitative methods to capture the complexity of the phenomena (Seaman, 1999). The research imperative therefore lies in finding an approach that makes the person explicit in the process and facilitates equally the discovery and application/validation of knowledge, through the disciplined observation of software architecture-based design in practice.

Based on the above argument, the research approach in this study must employ a situated method reflecting a real-world situation. Such an approach is congruent with the maturity stage of architecture-based analysis which sees it attempting to prove its practical application in broader fields in alignment with the external enhancement phase. Situated inquiry also simultaneously accommodates the world view of the research as well as addressing the 'validity' and 'evaluation' concerns of Shaw and Glass, raised above. Conducting the research in the rich environment of a complex real-world problem takes full advantage of the research situation. The importance of such a rich environment is exemplified by the emergence of the research concepts from the interaction of the researcher with the process of architecture-based analysis and the design group at large.

Where emphasis is placed upon the complexity of the real-world situation over and above that of theoretical deduction, research methods present a variety of modes of inquiry. The choice of approach is contingent upon the research aims as well as the nature of knowledge in the research situation. The following section discusses available methods of situated inquiry and the close relationship they share to particular theoretical perspectives.

3.2. *Situated inquiry and theoretical perspective*

The close relationship presented between research methods and specific theoretical perspectives highlights the importance of selecting a theoretical perspective as part of the research approach. Implicit in the selection of a specific theoretical perspective is a way of looking at the world and making sense of it (Crotty, 1998). This 'way of looking' will influence the relationships that are seen to exist in a situation and what is perceived as important knowledge. Consequently, the theoretical perspective is said to ground the assumptions associated with employing a specific research approach because it provides a rationale for the objects and methods of inquiry (Crotty, 1998). These methods of inquiry are typified by the three different research approaches of 'experimentation', 'ethnographic research' and 'critical research', which span the spectrum of theoretical perspectives from positivist, to interpretivist, to critical interpretivist[6]. While it would be unwise to suggest these research methods can only be informed by the theoretical perspectives presented below it is fair to say they are typical of research practice and clearly demonstrate the interdependency between tem.

3.2.1. Experimentation and Positivism

The methodology of experimental research is generally realised in situated inquiry through structured methods such as field experiments and hard case studies. Akin to experimentation on a subject that has human participants, methods like the hard case study strongly reflect the positivist ideals of accuracy and reliability which inherently beget repeatability. Ideals that have become embedded in the research approaches of traditional science to the point where scientific approaches to research are seen as the embodiment of logical positivism.

Arguing that there is no difference between the subject of inquiry being 'information rather than energy or matter', Tichy suggests that scientific method is equally well applied to the study of information processes as it is traditional scientific pursuits

[6] See Vidgen and Braa 1999 for a more comprehensive discussion of available methods of inquiry and their respective impositions on the research situation (Braa & Vidgen, 1999).

(Tichy, 1998). Enforcing the belief that knowledge is discovered and proved through the iterative application of a theory test and exploration approach. Tichy puts forward the view that experimentation can always be used, regardless of the amount of variables and potential inability to control them (Tichy, 1998). Techniques of experimental design such as control groups, random assignments and placebos ensure that the experimental process can always be applied. Even in the seemingly unbounded complexity of human subjects, *"The fact that a drug influences different people in different ways doesn't stop medical researchers from testing"* (Tichy, 1998, pp 36).

The distinguishing factor about the scientific approach is what is perceived as legitimate knowledge in the situation. Viewed as the 'tools of science' (Zelkowitz & Wallace, 1997), experimentation and data collection are required to distinguish between empirically verifiable knowledge and subjective knowledge in order to preserve the objective value of the finding (Crotty, 1998).

> *"Accept only that which is clear and distinct as true"* (Descartes in Hitchins, 1992, pp 13)

Positivism rejects human values, beliefs and judgements in much the same way as Galileo refused to accept attributes of things that were unquantifiable such as colour, taste and smell, preferring to focus on measurable properties such as size, shape and position (Husserl 1970 in Crotty, 1998). In order to tame the complexity of a real world situation, science makes useful abstractions of the characteristics of the situation (Crotty, 1998) to which repeatable methods of analysis, such as statistics (Tichy 1998), can be applied.

3.2.2. The Soft Case Study and Interpretivism

The soft case study tag is designed to refer to techniques such as ethnography and grounded theory, which have a strong grounding in the interpretivist theoretical perspective (Vidgen, 1997). The significance of such a perspective is again in how it views knowledge in the situation, whereas positivism rejected the unquantifiable *"secondary knowledge"* of any human situation, interpretivism *"looks for culturally derived and historically situated interpretations of the social life-world"* (Crotty, 1998).

Therefore soft case studies admit into the purview of their observations, elements noted earlier as incompatible with the scientific approach such as cultural and personal beliefs, values and ideals. As is the case in hermeneutics and texts, meaning is sought beyond their 'sheerly semantic significance' (Crotty, 1998).

Interpretivism proliferates the notion we are inherently social beings and that the meanings of our actions and words are deeply rooted in the interaction that takes place between ourselves and the objects of our world. Meaning is built from a cultural perspective and evolved through a personal one (Checkland & Holwell, 1998; Crotty, 1998). To some degree, like positivism, interpretivism seeks understanding. However interpretivism acknowledges understanding needs to be constructed by observing (empirically measuring) more than just externally quantifiable phenomena. Instead observations need to be interpreted in the context of social action (Crotty, 1998). A context that can only be fully understood from the individual perspectives of those involved. Hence the importance granted to unscientific elements of the situation such as beliefs and values. Consequently the validity of the generalisations (understanding) reached from the study are judged not by their statistical refutability, but by the logic and cogency of the reasoning used in describing the results (Walsham 1993 in Braa & Vidgen, 1999)

3.2.3. Action Research and Critical Inquiry

Critical Inquiry follows strongly in the interpretivist tradition of symbolic interactionism and hermeneutics, yet adds another significant dimension to the inquiry process, which is a necessary suspicion and disillusionment of existing social structures. Crotty suggests this dimension of criticism puts critical inquiry in stark contrast to interpretivism, stating the difference is between *"a research that seeks merely to understand and a research that challenges..........between a research that reads the situation in terms of interaction and community and a research that reads it in terms of conflict and oppression..........between a research that accepts the status quo and a research that seeks to bring about change"* (Crotty, 1998, pp 113)

In seeking to challenge our existing cultural understandings, critical inquiry entertains a phenomenological perspective. *"Phenomenology invites us to 'set aside all previous habits of thought, see through and break down the mental barriers which these habits* *have set along the horizons of our thinking'"* (Husserl 1931 in Crotty, 1998, pp 80). Yet if in promoting the setting aside and reconstruction of existing understanding phenomenology is suspicious of cultural meaning, critical inquiry is distrustful of it. Critical inquiry invites us to be suspicious of contemporary knowledge structures as potentially oppressive mechanisms of maintaining existing social order and institutions.

> *"Where most interpretivists today embrace such accounts as descriptions of authentic 'lived experience', critical researchers hear in them the voice of an inherited tradition and a prevailing culture"* (Crotty, 1998, pp 159)

The emancipatory tones are clear as critical inquiry seeks to do more than merely understand, it seeks to engage in social action with a view to realising equitable and idealised outcomes. It follows therefore that the primary methods applied with a perspective of critical inquiry are oriented towards invoking change in the situation in order to progress perceived problems and/or inequities. While some of these methods support more specific aims such as feminist research, others entertain less specific needs for social action such as Action Research (AR). Action research need only be initiated with a problematic situation requiring some degree of change. Extending the situated learning of symbolic interactionism, action research immerses the researcher in the problem situation (Checkland & Holwell, 1998) and permits learning through both reflection on and reflection in action (Sankaran, 2001; Schön, 1991). The learning is then used to guide further intervention in a continued spiral of action and reflection that so characterises action research (Dick in Sankaran, 2001).

3.3. Balancing knowledge interests and theoretical perspectives

In examining available research approaches and their associated knowledge aims, Vidgen noticed the similarity between these aims and Habermas' primary knowledge interests. Habermas' three knowledge interests of prediction, understanding and change align quite logically with the theoretical perspectives of positivism, interpretivism and critical inquiry, respectively. In both his thesis and subsequent publications Vidgen presents these three interests as forming the boundary points of a conceptual space of purposeful research activity in Information Systems (IS) (Braa & Vidgen, 1999; Vidgen, 1996). While information systems (IS) is a separate discipline to that of software engineering and computer science, it is still bound by the association with computing and technology. As such, it's research methods can be seen to a superset of those utilised within software engineering (Vessey et al., 2005).

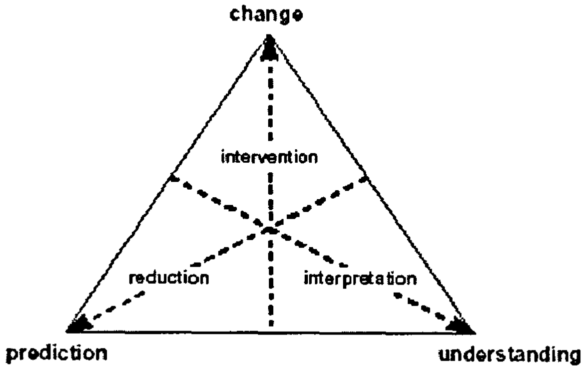

Figure 3.3 - An information systems (IS) research framework (Vidgen, 1996)

Although the work is acknowledged as being somewhat speculative (Vidgen, 1996), the framework is designed to illustrate the use of appropriate research methods to realise a specific knowledge aim. In the framework, techniques of reduction, interpretation and intervention are shown to achieve predictive power, understanding and change, which inherently inform the theoretical perspectives of positivism, interpretivism and critical inquiry. The boundary points are presented as unachievable ideals, suggesting the reality of research lies within the envelope of the framework. By locating the locus of research activity within the bounds of the framework, Vidgen aims to emphasise the point that research cannot simultaneously maximise all knowledge aims. Consequently research activity must trade-off ideals in the research situation.

Evidence for the trade-offs between research aims is prominent in the discussion of methods and perspectives in section 3.2. The scientific approach to situated inquiry spoke of the need to abstract the situation and provide repeatable results. However, as discussed in section 3.1 the abstraction and characterisation can impede developing a comprehensive enough understanding of the situation to yield results with real world application. Likewise the interpretive approach espouses the need to observe situated behaviours and record the rich tapestry of both quantitative and qualitative data available, yet it is not possible for the researcher to be involved as though they were indistinguishably part of the organisation (Vidgen, 1996). Conversely in seeking to effect change in the research situation, the critical approach depletes its capability to accurately document and understand the types of structures that un-critical interpretivism focuses on revealing.

Therefore in selecting a research approach consideration needs to be given to the trade-offs that will both benefit and detract from the research outcomes. The earlier discussion of research in software architecture in section 3.1 has already established the need for a situated method of inquiry. In section 3.2 three different theoretical perspectives and research approaches capable of facilitating such an inquiry were outlined and their trade-offs discussed with respect to knowledge interests. These characteristics can now be juxtaposed with the research situation and knowledge aims as described in chapters 1 and 2. A research approach is then formulated from the most appropriate research method and theoretical perspective.

3.4. Formulating a research approach

3.4.1. Selecting a research method

The first task in formulating a research approach is to select an appropriate research method. A theoretical perspective will help to justify the selection of a method as well as influence to some degree how the method is applied. However it is the consideration of how best to achieve the research aims given the research situation that drives research method selection (Crotty, 1998). In alerting the reader to the impact of the chosen methodology upon the structure of the thesis, chapter 1 has already presented some reasoning behind the adoption of a particular research method. This reasoning has been folded into the following discussion.

Chapter 1 established the research aim to understand and manage the manifestations of complexity within the architecture-based analysis process. Of particular interest was the notion of 'people' complexity stemming from early interactions within the design group. The nature of 'people' complexity is inherently idiosyncratic and dependant upon the individuals within the situation. Selecting a situated method of inquiry addresses a practical project need to analyse architectures effectively and a research need to better understand the situational complexity of the analysis process. In doing so the research also addresses a primary issue of relevance to architecture-based analysis. Both the maturity of architecture-based analysis as a technology and as a research discipline sees it attempting to further validate its processes in broader real world contexts.

The idea of a situated method of inquiry serving both a practical project need and a research need implicitly raises the issue of role duality. In research collaborations between industry and academia, there is an inherent need to fulfil the research objectives of the industrial partner as well as the personal research aims of the researcher. As part of the project cohort the researcher has a responsibility to research the design of an effective next-generation network (NGN) management system. As a personal researcher the responsibility rests with contributing to knowledge on architecture-based analysis. Action research (AR) as a methodology for situated inquiry is sympathetic of the need to perform both roles, valuing the insight gained through participating in the same process which is being researched.

Designed around participation and intervention in problem situations, action research is also accepting of change as a mechanism of both developing further understanding and improving the problem situation. As a result action research informs both theory and practice, which both creates an ideal learning scenario for an applied discipline and addresses Shaw's concerns for validity in existing architecture-based design research. Furthermore the ability to use the insight to effect change where there is a perceived problem supports the primary role of the linkage researcher, which is ensuring they participate effectively to achieve the industry partner's research objectives. Participating effectively in this instance is highly likely to call upon change as a means of progressing. Chapters 1 and 2 have presented on a significant aspect of complexity in the analysis process that doesn't appear to be managed by current techniques. The implication is that unmanaged complexity is likely to impede the analysis process engendering change in order to ensure the intended analysis outcomes.

Finally, and importantly for the consideration of the theoretical perspective, action research has a strong history in the interpretivist tradition. Action research admits into the scope of concern key aspects of complexity, which are not considered legitimate subjects of study in the view of traditional science. Developing understanding about these aspects of complexity is a key part of the research aim. The following discussion will elucidate the importance of this association between action research and interpretivism in addressing the importance of the theoretical perspective to the research situation.

3.4.2. Seeking a theoretical perspective

In discussing the recent development of architecture-based design principles, Chapter 2 noted the recent integration of architecture-based analysis into the design lifecycle. The closer integration is deigned to accommodate iterative incremental approaches preferred in complex design situations. In doing so methods of architecture-based analysis have evolved from expert centric and evaluation focused to stakeholder centric and analysis focused. The consequence of expanding the stakeholder group was then expressed in terms of opening up a dimension of complexity referred to as the 'people' dimension by Flood (Flood, 1988). Whereas the traditional 'systems dimension' to complexity is concerned with the number of components and relationships comprising a system, the people dimension incorporates the capabilities, notions and perceptions of the people involved. The research aim clearly expressed the need to understand this dimension of complexity working from the preceding argument that it had not been adequately addressed in research thus far.

The emphasis on understanding how aspects of complexity, beyond the systems dimension, impact the process of architecture-based analysis appears to strike an immediate accord with the interpretivist tradition discussed in section 3.2.2. Interpretivism acknowledges the significance of artefacts extends beyond their extant form and should be viewed as a product of the personal beliefs, values and ideals of the creator. Similarly the interpretation of an artefact will involve the interaction between a cultural perspective shaped by the afore-mentioned elements of the reader's object world and the artefact itself. The incorporation of perceptions, values, beliefs and ideals into what is considered valid knowledge in the research situation shows an inherent compatibility between the aims of the research and the interpretivist theoretical perspective.

Conversely the validity granted to elements of the research situation such as those in the interpretivist tradition sees the research begin to distance itself from the positivist viewpoint. The need for replication (repeatability) in scientific inquiry ensures that knowledge of a subjective manner is jettisoned because of its inability to be measured and empirically verified. The types of knowledge that science looks for in situated inquiry goes only so far as that which allows causal attributions to be made, honouring the predictive ideals of positivist research. If we accept the interpretivist viewpoint then the empirically verifiable facets of the research situation are unlikely to fully explain the process and its outcomes, especially with regard to culturally-based phenomena which are seen as resistant to explanation in causal terms (Crotty, 1998).

> *"....characteristic of the human sciences....They accept that one can satisfactorily understand the natural world simply by understanding the parts that make it up. In the case of human sciences, however, this simply will not do."*(Crotty, 1998, pp 92)

Aspects are likely to emerge from within the process due to the scope of human participation that can only be attributed to understanding the research situation at hand. These emergent properties mean the results cannot be immediately abstracted to some broad general theory. The focus on the uniqueness of the situation at hand reveals the ideographic nature of interpretivism versus the nomothetic nature of positivism (Crotty, 1998). Whereby nomothetic entails a focus on regularities (laws) and ideographic focuses on the individual. Taking a scientific approach to examining architecture-based analysis techniques implies experimentation can provide adequate explanation to fulfil the research aim. However chapter 1 and 2 simply established the scope for extended dimensions of complexity in architecture-based analysis, it did not purport to predict or theorise their manifestations. Experimentation would require that the architecture-based analysis process be divided into empirically quantifiable phenomena, theories devised about the affects of those phenomena on the process and finally careful control exerted over all but one variable in a bid to show how its manipulation affects the process. The dismissal of important subjective elements aside, the sheer number of variables in the process and the uncertainty about their impact would make such an approach extremely difficult.

Still the scientific tradition echoed largely in a lot of the computer systems literature maintains that such complexity is not beyond the capability of experimental methods. In situations where the number of experimental variables is extremely large, and in many instances out of the control of the researcher, science has developed methods of abstracting out the essential detail and seeking meaning in statistical and other legitimate methods of generalisation. Tichy's earlier statement about experimentation with drugs despite variability in human reactions is highly representative of this viewpoint. Where understanding is incomplete, the situation is overcome by tightly controlling and regulating a singular variable of the research situation, revealing its affects on the system and paving the way for more structured experimentation to follow (Zelkowitz & Wallace, 1997). However section 3.1 already remarked how the abstraction of detail used to create an externally valid research setting has encouraged research in the field of software architecture that rarely addresses the complexity of the motivating real-world problem, highlighting the *"artificiality of splitting out single behavioural elements from an integrated system"* (Foster 1972 in Checkland & Holwell, 1998, pp 22). In the context of the current research project it is the complexity of the real-world situation that sparked the research aims and it is indeed a complex reality within which the research will be conducted.

The preceding discussion has demonstrated the alignment of the interpretive viewpoint with what is considered important knowledge in the research situation. It has also discussed how scientific experimentation, seen as the implementation of positivism is unlikely to fulfil the research aims. It is with some confidence that the selection of an interpretive perspective can be adopted for the research project. Discussion now needs to focus on whether a purely interpretive or critical perspective should be adopted. Until now the consideration of the critical perspective has been embedded in the discussion of interpretivism. The justification being that critical inquiry accepts the interpretivist view of what is important knowledge in the situation, yet offers a significantly different view of how such knowledge should be interpreted and used. The following section will discuss this difference between the two and its consequence for the research method.

3.4.3. Interpretive and Critical aspects of action research

The framework of Figure 3.3 suggests a differentiation needs to be made between the values of change and understanding when deciding upon an appropriate interpretive method of inquiry. The difficulty with the proposed research is that a measure of both change and understanding are desired. Change is required because perceived shortcomings in existing analysis techniques are likely to impact the progress of the architecture-based design, which relies heavily on the iterative application of evaluation. In the event that issues prevent the outcomes expected of the evaluation, some agility is essential in order to continue the project work. Additionally the research aims as expressed in chapter 1 coupled with the discussion of emergent learning through situated inquiry demonstrates the need for understanding.

At first the situation presents itself as requiring a trade-off between understanding and change, with possibly a multi-method or tailored approach necessary. However utilising the framework in such a manner provides insight into the potential misconceptions which can grow from rigidly interpreting the axes and boundary points of the framework. It also highlights the danger in mapping research methods to theoretical perspectives. Although there are typical trends and the theoretical perspective will influence the manner in which the research method is used, the two elements are not necessarily synonymous (Crotty, 1998; Klein & Myers, 1999).

Action Research by definition invokes change within the situation, however the action within the situation is designed to create further understanding as a parallel outcome (Dick in Sankaran, 2001). Hence action research places equal importance upon both the stages of change and understanding, rather than a single minded focus on change. What does occur is that the situational understanding becomes somewhat incomplete due to the constant flux in the situation. However to some degree this is exchanged for knowledge on how change within the situation affects the current framework of knowledge. This adaptive knowledge, often regarded as responsiveness[7] is seen as critical because of the perception that something in the situation is 'wrong' and needs to be changed to facilitate progress.

The notion of change aside, action research has also been used as a vehicle for primarily creating situational understanding. Through commonly applying the action research method within organisational systems research, the discipline of information systems is acutely aware that large scale change is often unrealisable. Power structures and commercial imperatives within organisations commonly prevent meaningful change (Braa & Vidgen, 1999). Quite distant from the traditionally emancipatory action

[7] http://www.scu.edu.au/schools/gcm/ar/art/arthesis.html

research literature, information systems (IS) puts faith in the value of action research as a grounded learning mechanism, whereby *"research informs practice and practice informs research synergistically"* (Avison et al., 1999, pp 94). In his thesis Richard Vidgen presents the 'action case' methodology, which is designed to equally balance the understanding and change research imperatives. The method he employs to do so is based upon Susman's AR model (Susman & Evered, 1978). Dick offers further corollary of the action research as a method of learning by suggesting action research takes place across a spectrum of concerns, the first of which is understanding, the second is practical, the final of which is critical research (Zuber-Skerritt in Sankaran, 2001).Final evidence of the synergy between action research and the interpretive tradition, lies within the hermeneutic circle, which mimics the cyclical, reflective nature of AR. The hermeneutic circle initially presupposes a *"rudimentary understanding of what one is trying to understand"*, akin to understanding the whole by grasping at its parts. Understanding is then developed by *"divining the whole"* and returning to *"illuminate and enlarge one's starting point"* (Crotty, 1998, pp 92).

From this it is evident that the critical perspective does not necessarily entail the hefty burden of helping to realise a utopia of social justice or equity as is the case with feminism and Marxism. Some distrust in existing methods of architecture-based analysis is implied by the research aims but it is difficult to perceive their current shortcomings as culturally influenced by social structures designed to oppress and preserve design activity within its current bounds. Alternatively a critical perspective is offered that modestly values responsiveness in the situation in order to overcome perceived difficulties and problems. A perspective that is inherently focused on understanding but acutely aware of the potential for change to improve the situation. It is also a perspective that has shown to comfortably integrate with the application of the action research method of inquiry.

Therefore the research approach can be said to incorporate the situated method of inquiry known as action research, applied with a largely interpretive theoretical perspective that is critical by necessity (practical in Dick's terms 7). Having derived the theoretical framework underpinning the research, the following section will grant body to the structure and use of action research, describing in more detail how it will be instantiated with this research project.

3.5. Applying action research

3.5.1. Origins

Growing out of post-positivist philosophy of science (Baskerville & Pries-Heje, 1999), action research (AR) is a methodology for inquiry into human activity systems. Action research is derived according to the principle that the social world is not governed by the same physical regularities of the universe which traditional scientific method seeks to uncover. Instead research in these social contexts *"becomes as organised discovery of how human agents make sense of their perceived worlds, and how those perceptions change over time and differ from one person or group to another"* (Checkland & Holwell, 1998, pp 22).

The focus on complex social systems derives from its origins with the fields of operational research and social psychology, amidst the social upheaval precipitated by Wold War II (Baskerville & Wood-Harper, 1996). The methodology has since established itself as the dominant paradigm for the practice of organisational development (Baskerville & Wood-Harper, 1996), as well as finding use within other socially applied disciplines such as Education (Dick, 1993), Welfare (Coghlan & Brannick, 2001) and Health (Checkland & Holwell, 1998).

The strong historical precedence within complex social systems may appear to initially disparage efforts to apply the action research paradigm within areas of less cultural and social significance. However the binding facet of social systems is that they are all operated upon by what can be termed 'applied disciplines'. Applied meaning there is some unavoidable 'clinical' aspect to the professions such as organisational development, education, etc. Without the application of these fields to real social systems, there is no field. The social nature is inherent in being an applied discipline. The earlier discussion of software architecture in section 3.1 established quite clearly its credentials as an applied discipline, developed from the field of practice. Software engineering is clearly concerned with making things, the success of which should be judged by the success of the user making use of the creation (Brooks, 1996).

Action research has a broad relevance as revealed within the structure of its cycle which resembles that of experiential learning itself (Dick in Sankaran, 2001). As a result action research can be used by any professional in an applied field as a means of furthering learning within their discipline. Beginning with the application of theory to a real world situation, followed by critical reflection of the outcomes, action research creates a cycle in which theory informs practice and practice subsequently informs theory (Avison et al., 1999). Schon's 'reflective practitioner' exemplifies such a cycle of professional development based on intervention and reflection (Schön, 1991).

3.5.2. Structure

As its name suggests action research is organised around phases of understanding and change, or as Blum proposed in 1955 a 'diagnostic stage' and a 'therapeutic stage' (Blum, 1955). A key aspect of the process as it is commonly portrayed is the cycling from one phase to the next. Change is always incremental and the effects of change can always be used to inform theory, which once altered may suggest evolved action. Social systems are dissimilar to structural entities in that they shouldn't necessarily be perceived as having problems to be solved, but rather situations to be managed. The subtle but important difference is that once managed, the situation does not go away. Time, perceptions, goals, can all change requiring constant vigilance. Therefore the spiral of action research is potentially infinite, though in reality it of course does conclude for reasons specific to each situation (time , money restrictions, understanding achieved, satisfaction achieved, etc). Figure 3.4 below, depicts the staged, cyclic nature of action research, where the action and reflection stages are analogous to the therapeutic and diagnostic stages mentioned above.

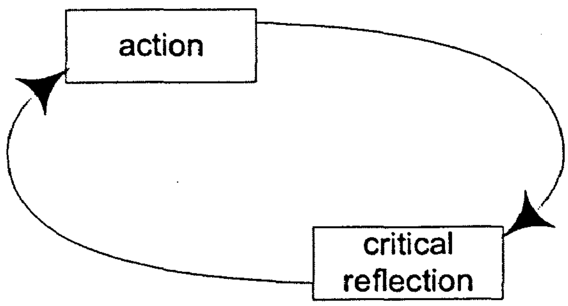

Figure 3.4 - Dick's description of AR as cycling between action and critical reflection (Dick in Sankaran, 2001)

Such a model is of course an abstraction of what takes place in reality, which is always more complex and planned. Therefore, seemingly in response to the scientific grumblings about its lack of rigour, the action research community has sought to further expound the process (Baskerville & Pries-Heje, 1999). Ensuring that the research process is at least consistent, granted the research situation can never be. One of the more commonly referenced models of action research presents it as a *"cyclical process with five phases: diagnosing, action planning, action taking, evaluating and specifying learning. The infra-structure within the client system and the researcher maintain and regulate some or all of these five phases jointly"* (Susman & Evered, 1978, pp 588).

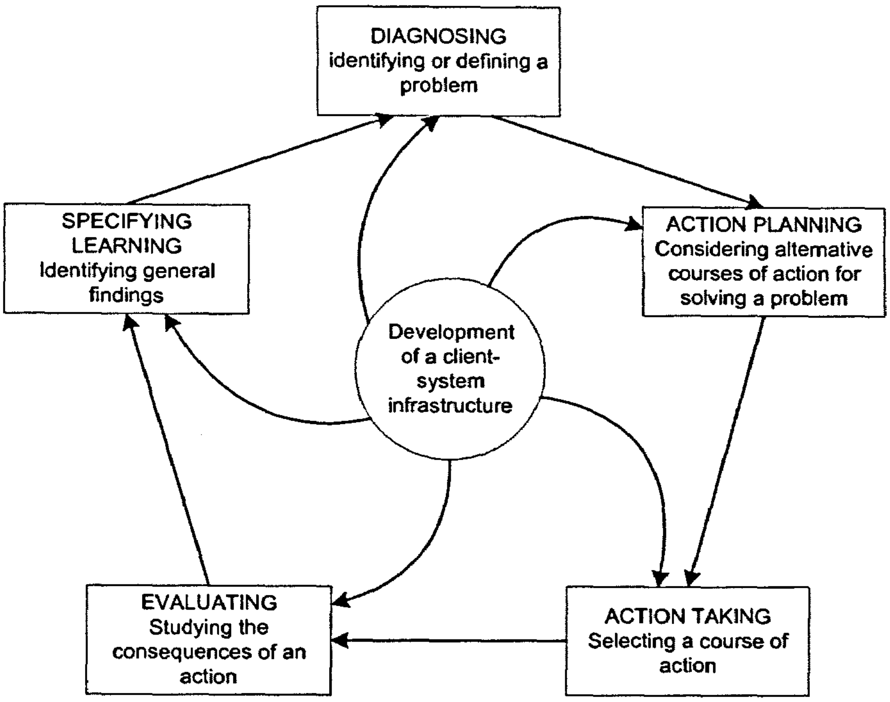

Figure 3.5 - Susman and Evered model of action research (AR) (Susman & Evered, 1978)

Whilst the model of Figure 3.5 elaborates on the action research process itself, it remains prescriptive of stages and says nothing of what happens to knowledge in the situation. Action research commonly comes under fire from proponents of an experimental approach to inquiry, because it does not adequately account for learning in the process. Science begins with structured knowledge, a hypothesis about a variable, a method in which that variable is to be manipulated and a plan for how measurement should be conducted. Action research can begin with a sense of something wrong, and action and theory altered within the structure of the cycles. Without dealing explicitly with what is considered knowledge in the situation, action research opens itself up to criticism as being merely 'consulting' (Baskerville & Wood-Harper, 1996) and literature reporting its application as 'no more than anecdotes' (West & Stansfield, 2001). In reaction to such criticism Checkland proposes that the use of an explicit, 'declared in advance' framework of knowledge is an essential element of a general research model (Figure 3.6) (Checkland & Holwell, 1998) and should be implemented by all action researchers. Derived from early experiences encouraging research students to explicitly consider their theoretical stance, the framework presents the theoretical basis of the work with regard theory in relevant disciplines (Checkland, 1995).

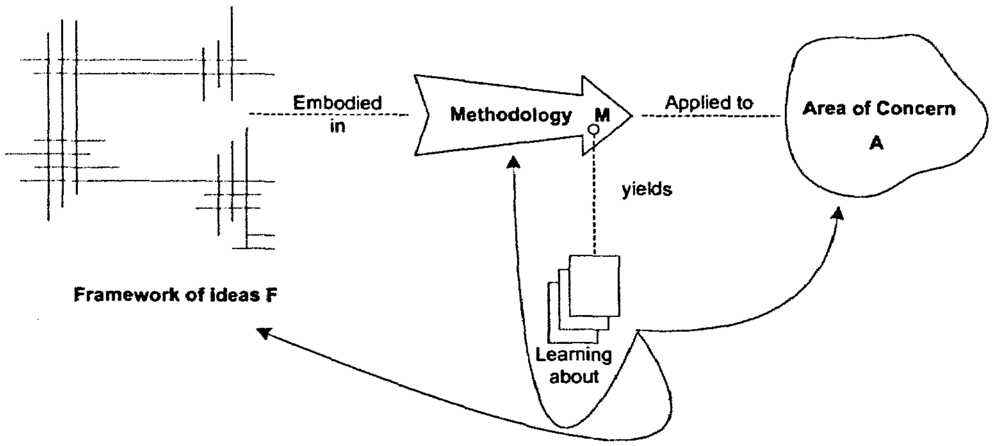

Figure 3.6 - Elements of a generic research process (Checkland 1998)

With the addition of a framework component, the general research model aptly describes the research process of science, equally as it does interpretive inquiry (Checkland & Holwell, 1998). The framework is the critical mechanism through which knowledge in the situation is both declared and altered. Therefore learning is able to be specified with respect to the framework, in much the same way as reflection upon the problem situation and the research approach can be specified with respect to the methodology (M) and the area of concern (A). Consequently the changes to knowledge in the situation become transparent within the writing, addressing the calls for more open practices.

The process model of action research resulting from Checkland's general research model (Figure 3.7) exhibits most of the steps shown in that of Susman's Figure 3.5.

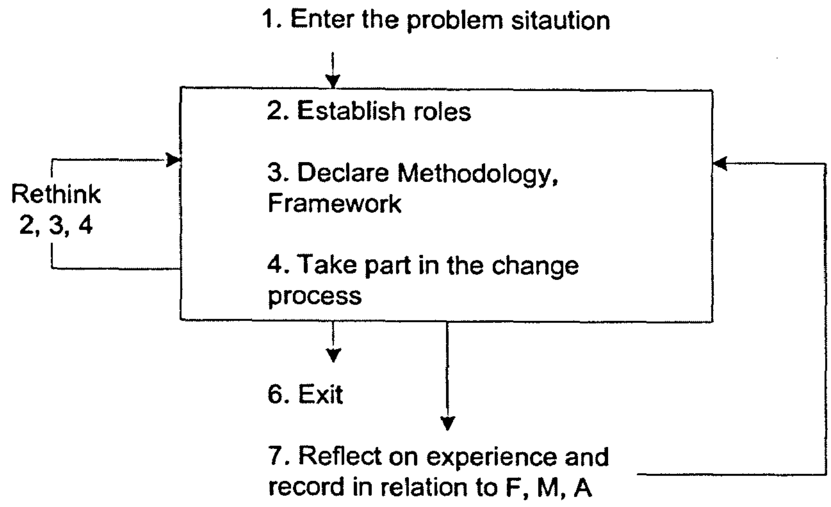

Figure 3.7 - An Action Research methodology incorporating an explicit intellectual framework (Checkland & Holwell, 1998)

Both represent the need to iteratively implement change within a situation and reflect upon the results with respect to established beliefs. However in terms of providing structure to action, the process model of Susman appears to describe in a more detailed fashion the steps that comprise action and subsequent reflection. In the interests of transparency, it appears wise to preserve the clarity of Susman's model, whilst incorporating the framework elements of Checkland's. Consequently Figure 3.5 and Figure 3.6 will provide the backbone of how action research is practically applied within this research project.

3.5.3. An initial research infrastructure

As represented in both Susman's and Checkland's models (Figure 3.5 and Figure 3.7), the first step in an action research project is diagnose a situation as problematic. Following this the researcher needs to establish a client system infrastructure, which incorporates the negotiation of respective stakeholder roles in the research situation. The infrastructure of the collaborative project can be best described by outlining the relationship between the participating groups and the role of the researcher within that.

The industry-academic collaborative research and development initiative seeks to align the research and development needs of an industry partner with the expertise of university research groups. In this instance the project targeted the utilisation of open systems and architecture-based design principles to better understand and design a policy-based next-generation network management system. The university research team consisted of academics with mixed research and industry experience in the areas of telecommunications and computer-based systems engineering. The group was essentially peered, though formal roles existed in the form of a chief investigator who was responsible for the administration of the project, principal investigators, a post-doctoral fellow and doctoral research students. The industry partner participants were all considered principal investigators as well. The researcher's supervisors were the post-doctoral fellow, the chief investigator and one industry supervisor. The research was divided into core research disciplines including policy-based management, network simulation and implementation and systems architecture. The researcher's main tasks were within the discipline of systems architecture, though work within the other areas was of course necessitated by understanding.

Therefore the role of the researcher was, along with the chief architect, to guide and develop an architecture-based design process that would produce a proof of concept architecture for the management system. Of course in action research, the researcher bears the dual responsibility of acting within the situation and acting on the situation. The two roles here are referred to as researcher and participant. The role of participant saw the researcher answerable to the group for their progress and contribution to the development of an architecture. The role of researcher saw them answerable to their supervisor and ultimately the greater research community for the quality of their research.

Having established the research situation, the infrastructure of the research and the roles within it, the next step is to plan and take action. Guidance for how to begin is evident in Checkland's process which promotes the need to establish the framework of knowledge (F) and methodology of intervention (M). The area of concern is naturally defined by entry into the problem situation, in this case the design of a management framework for a next-generation network in a linkage research project. The remaining (F) and (M) elements are presented in the following chapter due to their close association with the actual action research intervention.

Summary

In developing an initial research framework chapter three began by examining the current state of affairs in computer-based systems research, with a particular focus on software architecture. The discussion revealed the bias of existing research towards formulating and proving theories in a predominantly analytical way. Such a focus reveals a challenge which is to prove the application of such methods to the complexity of real-world problems, something seen as largely unaddressed in both computer systems and software architecture research alike. Architecture-based analysis was shown to have matured to the point where it is now attempting to validate its work through broader application to real world design situations. Consequently the research aims and current stage of maturity of the process were seen as promoting the use of a situated method of inquiry.

Some common methods of situated inquiry were subsequently presented along with a discussion of the theoretical perspectives that such research typically aligned itself with. The consequences of these typical methods of inquiry for the knowledge aims of the research were then discussed. Following on from this, section 3.4 reasoned through the selection of a research method and theoretical perspective. The ability for action research to entertain a dual project-researcher role as well as its acceptance of change saw it selected as the most appropriate research method. The alignment of both action research and the knowledge aims of the research with the interpretive tradition resulted in a largely interpretive theoretical perspective that was critical by necessity. Chapter 3 concluded by presenting a model of action research combining the stages of Susman's model with the research process model of Checkland.

Detailing the research approach concludes the first phase of this thesis. Illustrated in Figure 1 of chapter 1, the first research phase is aimed at identifying the research to be conducted, describing its practical and theoretical contexts (situating) and the way in which it will be conducted (planning). Consequently chapter 4 will describe the first stage in the actual implementation of the research, presenting the first action research cycle exploring the use of architecture analysis as a design learning technique.

4

Design learning and the ATAM

Introduction

Chapter 1 introduced the research situation of the collaborative design of a network management solution. Motivated by the application of architecture-based design principles to complex design problems, the project brought together industry and academic stakeholders. Chapter 2 then discussed the research focus of architecture-based analysis, presenting an evolved view on the complexity of the task. This evolved view incorporated a greater concern for complexity arising from individual stakeholder perspectives. The need to further understand the effect of this complexity on the process and its outcomes was clearly stated, echoing the research objective stated in Chapter 1. Taking the intended research aims and the research situation into account, Chapter 3 derived critical inquiry and action research as the most appropriate elements of a research approach.

Chapter 4 combines the theoretical, methodological and situational elements by beginning the account of the learning and experiences that took place during the first action research cycle. The first section will focus on formally establishing a framework of ideas. Following the presentation of the framework, consideration is given to the selection of an appropriate method of architecture-based analysis. The Architecture

Tradeoff Analysis Method (ATAM) is selected and introduced. The remainder of chapter 4 presents the first application of architecture-based analysis. The structure of chapter 4 is shown in Figure 4.1.

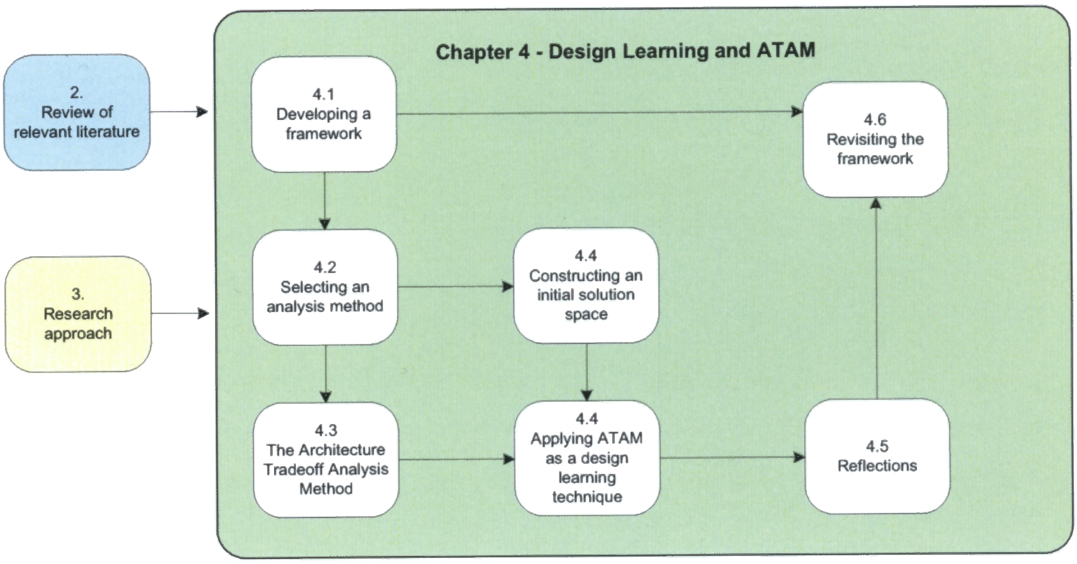

Figure 4.1 – Structure of Chapter 4

4.1.Developing a framework

As discussed in the research methodology (section 3.5), if action research is to be perceived by its audience as more than mere 'novel writing', there has to be a transparent presentation of an intellectual framework of ideas *"in terms of which what constitutes 'knowledge' about the situation researched will be defined and expressed"* (Checkland & Holwell, 1998, pp 22-23). The relationship between what is considered knowledge in the situation and the framework of ideas exposes it as an episteme, effectively a neatly abstracted set of linked ideas within which knowledge in the research situation can be defined.

Entering the action research project with a defined framework is invaluable in applying a research method which is both iterative and naturally engenders change. Amidst the change, the framework provides a frame of reference from which to 'reflect, judge and move forward' (West & Stansfield, 2001). Such a frame of reference assists both the researcher in maintaining awareness of the development of their ideas (West

& Stansfield, 2001), and the research audience in being able to recover the process by which the results were obtained (Checkland & Holwell, 1998).

The theoretical framework of ideas underpinning this research has been presented in the first two chapters. At the heart of this initial framework was the idea that non-technical issues could be determining factors in analysis outcomes. Discussion focused on the literature precedent of 'social, psychological and managerial issues' affecting the analysis. Insight into these issues was offered by adopting a softer view of the architecture-based analysis process derived from principles in broader design-related research disciplines. Principles like Weltanshauungen (W) and object worlds acknowledge the individual viewpoints brought to the design situation by each participant. The net effect of these viewpoints is heightened situational complexity, which Flood suggests is comprised of a 'systems' dimension and a 'people' dimension, where the 'people' dimension is a result of personal beliefs, perceptions and capabilities (Flood, 1988).

In acting as a medium through which the stakeholders communicate in a design situation, the interpretation of design artefacts is naturally influenced by these object worlds which are unique to each participant. The resultant mental filter through which each participant views reality suggests they don't share a common view or understanding of the system at hand (Galle, 1999). Consequently understanding needs to be fostered through the explicit negotiation of meaning, an aspect that is commonly overlooked amidst the need to resolve more explicit design issues such as system goals and function. Further confounding attempts at negotiating meaning is the nature of architectural representations, which inherently can be quite abstract in the early stages of the design lifecycle. Refuge to such uncertainty in design situations is often sought in the requirements artefacts. However these requirements artefacts are commonly presented as problematic due to the fact they are seldom well defined, particularly in complex problem situations.

The distillation of these concepts from the broader theoretical work helps to articulate the essential concepts that have had a significant bearing on the research undertaking. The collection of these ideas with respect to the problem situation can be seen to form the initial framework for this research undertaking, as depicted in Figure 4.2.

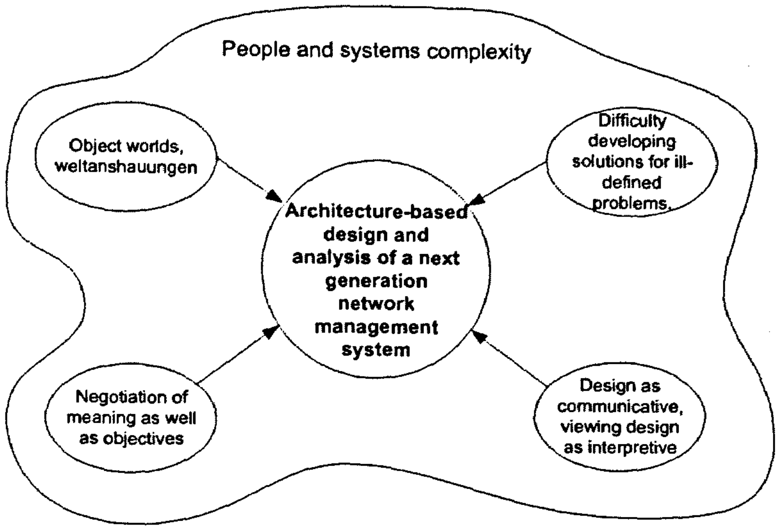

Figure 4.2 - A framework understanding the complexity of architecture-based analysis

Utilising the framework of Figure 4.2 in the process of investigating the architecture-based design and analysis of a next-generation network management system realises the instantiated version of Checkland's research model depicted in Figure 4.3.

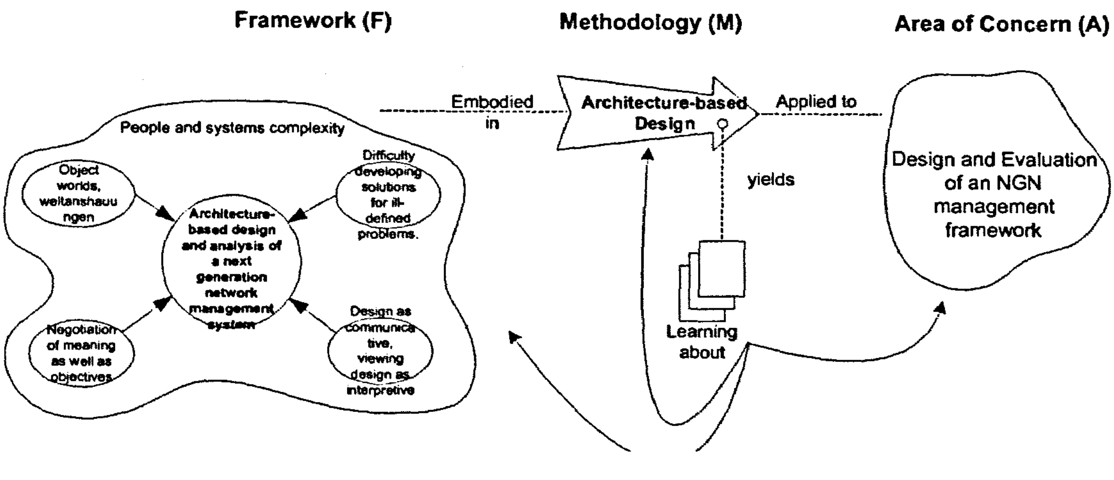

Figure 4.3 - FMA Research Model

As discussed in Chapter 1 and declared in Chapter 3, the area of concern (A) is the design of a proof of concept framework for a NGN management system. The focus on software architecture design principles as being beneficial to the design situation sees architecture-based design employed as the methodology (M). Shown in Figure 1 of Chapter 2, the architecture-based design process borrows the iterative incremental aspects of Boehm's spiral life-cycle model of software development (Boehm, 1988) and augments it with a quality focus (Bosch, 2003). The methodology of architecture-based design (ABD) (Bachmann et al., 2000) may seem broad granted the theoretical focus on the specific area of architecture-based analysis. However the methodology of design, rather than a more specific analysis-oriented methodology is vehicle through which the research will be conducted. Architecture-based design principles see architecture-based analysis as being applied iteratively throughout the stages of design from early conceptual artefacts through to the more refined specifications of a system (Bachmann et al., 2000). As such the methodology of architecture-based design provides a basis for the application of architecture-based analysis in an overall process. This proves a more faithful model of the research since the intervention is based on the researcher's study of and participation in, the design process of an NGN management system, as opposed to arbitrary interventions utilising architecture-based analysis.

The final element of the research situation that needs to be resolved is the method of architecture-based analysis that will be used throughout the design process. The following section discusses the available methods with respect to the research situation.

4.2. Selecting an Analysis Method

Chapter 2 presented the methods of architecture-based analysis as falling into one of two categories, questioning or measuring. The most significant difference between the two categories is that questioning methods offer qualitative results and measuring methods offer quantitative results. In complex design situations the effort required to develop models suitable for quantitative analysis tend to dissuade the use of measuring techniques. The subsequent measurements used to evaluate the architecture also tend to be focused on specific aspects, such as performance and are not broadly applicable to other qualities (Abowd et al., 1997). Conversely, questioning techniques provide a comparatively resource efficient means of analysing the architecture for any quality attribute. These characteristics of low resource cost and broad coverage suggest questioning analysis methods could be applied incrementally throughout the design lifecycle. However for reasons discussed in section 2.2 of Chapter 2, the applicability of these methods extends only to selecting between candidate solutions once the design has reached maturity (Lee & Hsu, 2002).

The adoption of an iterative incremental development process for the collaborative NGN design project required a method which could be used throughout the systems development lifecycle. It was an important criterion that the analyses provide insight into design issues and how they relate to the customer objectives. Together with the need to cover multiple quality concerns and use as little resources as practicable, these requirements formed the main criteria for selecting an analysis method. Methods offering these characteristics include Software Architecture Assessment using Bayesian Networks (SAABNet) (van Gurp, 2000), Software Architecture Review and Assessment (SARA) (Obbink et al., 2002) and the Architecture Tradeoff Analysis Method (ATAM) (Kazman et al., 2000). Although it is viewed as qualitative in nature, SAAbNet requires the numeric coding of relationships between design aspects as conditional probabilities. Defining the relationship between design moves and subsequent system properties is still being researched in architecture. Let alone knowing the strength of the relationship and the degree to which change in one changes the other. Such determinism requires the type of foreknowledge that is unlikely to exist for sometime in architecture-based design.

Determinism and coding are neither traits of ATAM nor SARA which both focus on the relationship between architectural strategies and key system requirements. Parallels are strong between the use of scenarios and sensitivity points within ATAM and the use of Architecturally Significant Requirements (ASR)s and Architecturally Significant Decisions (ASD)s within SARA. Additionally the activities outlined in SARA as well as the goals and outcomes are all comparable in nature to that of ATAM. Taking into account their similarities, ATAM tends to stand out over SARA because it is a comparably mature and tested evaluation method. ATAM has evolved from an earlier method of SAAM, itself considered relatively mature (Dobrica & Niemela, 2002) and has since been applied and reported in several case examples. Therefore given the project at hand ATAM presented itself as the most appropriate method of architecture-based analysis.

4.3. The Architecture Tradeoff Analysis Method (ATAM)

The Architecture Tradeoff Analysis Method (ATAM) is a questioning method of analysis developed within the Software Engineering Institute (SEI)[8]. The ATAM implements the scenario-based analysis of its predecessor, the Software Architecture Analysis Method (SAAM) (Kazman et al., 1994). Scenarios are structured so they describe the stimulus exciting the system into use and the subsequent behavioural response of the system. The scenarios effectively provide a qualitative test of the architecture. The use of scenarios allow multiple quality requirements to be examined at the same time, as scenarios can be devised that target a specific quality (Abowd et al., 1997).

The main objective of the ATAM is to identify and prioritise areas of risk within the architecture that should be granted greater attention during the analysis, design and implementation phases. Risk can arise from architectural decisions that are deemed to have not been made, yet are important to the system behaviour. Aspects of the design that are important to the behaviour associated with more than one quality are also noted as 'trade-off' points. These trade-off points are also considered architectural risks. The focus on risks is based on the belief competing goals are inherent in design. A focus on one goal at the expense of another without first validating the design choice with a stakeholder is highly likely to lead to an unsatisfactory solution. The ATAM provides a ready vehicle for discovering these issues throughout the design lifecycle and involves key stakeholders in the process. Involving the stakeholders grants them visibility of the decisions and enables their views to be expressed in the process.

The ATAM consists of two phases, where the second phase repeats the activities of the first phase and conducts 3 steps unique to the second phase. The first phase is designed to familiarise the parties with the process and is conducted in a consulting manner in order to grant scope for developing and gathering important analysis artefacts. The first phase involves stakeholders designed to represent the three key groups, the facilitators, the client and the solution architects. This first phase group is a subset of the overall stakeholder group. The second phase draws in an expanded group of

---

[8] http://www.sei.cmu.edu/

stakeholders and expands on the artefacts developed in the first phase. The phases and stages of the ATAM are depicted in Figure 4.4 and described below.

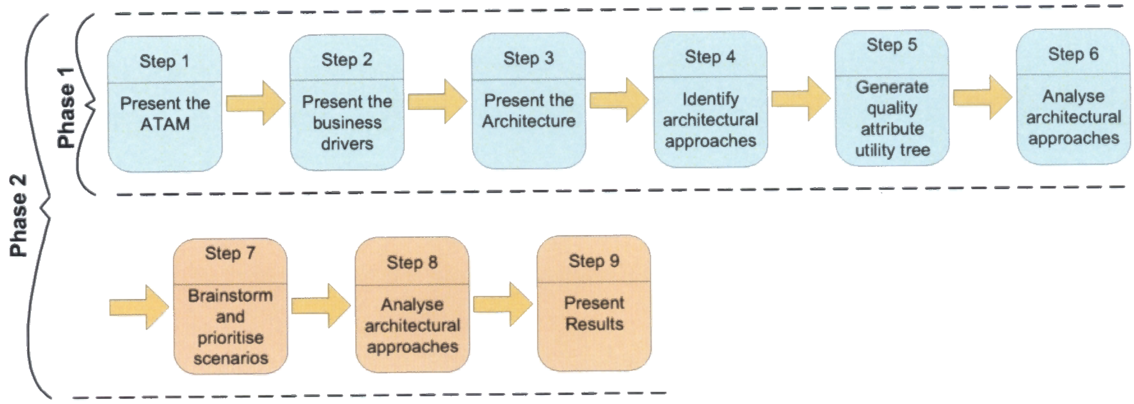

Figure 4.4 - ATAM Structure

The first step introduces the method itself to the participants. This is followed by two steps presenting both the business case and the solution architecture respectively. The 4th step looks to identify key architectural approaches responsible for system qualities. The 5th step creates the attribute utility tree, which refines the business drivers into quality goals into concrete scenarios representative of the goals. The final steps identify architectural sensitivity (architectural decision key to a specific quality) points, trade-offs (architectural decision key multiple qualities) and risks (important decisions not made)

Having set in the place all of the elements that comprise the research setting, its theoretical underpinnings as well as the methodologies guiding both the research into and the application of architecture-based analysis, the following section will begin the account of situated learning that took place in this action research project.

Having set in place all of the elements that describe the research setting such as the framework of ideas, methodology of intervention and the situation itself, the remainder of this chapter will begin the account of situated learning that took place in this action research project. The next two sections establish the project context surrounding the first use of the ATAM. Constructing an initial solution space examines the primary challenges early on in a project and how design activity is organised around this. Applying architecture as a design learning technique discusses how analysing existing architectures significantly contributes this initial activity.

4.4. Constructing an initial solution space

As is commonly the case in design, the generation of an initial solution space begins with a review of existing products and literature as well as other system types with which a strong analogy can be drawn (Joseph, 1996). This activity, described by Schon as 'problem-setting activity', inquires into the situation in order to understand what is unique about it. Problem-setting also seeks to establish its similarity to other situations so that the familiar aspects may *"function as a precedent, or a metaphor, or-in Thomas Kuhn's phrase-an exemplar for the unfamiliar one"* (Kuhn 1977 in Schön, 1991, pp 138).

The brief given to the collaboration was intentionally vague so as not to constrain the creativity of the design approach. The main elements of the design challenge confronting the group comprised of the need to develop a management system framework for a next-generation network (NGN), using the principals of open systems and a popular industry archetype of policy-based architectures. Due to the prominence of the telecommunications dilemma and the world-wide applicability of a solution, a considerable amount of both commercial and academic material existed in the area of NGN management. Initial search activity brought to light many existing approaches and theories on how to create an effective NGN management framework.

On examining these approaches in the design meetings it became evident that despite the more common need to deliver new telecommunications service types over a carrier network, a great deal of diversity existed within the solution architectures. Of interest to the group was the fact that different design approaches did not seem to solely account for the diversity. Differences could also be seen to arise from the facet of the NGN problem motivating the design approach. Some designs focused on particular aspects of networking key to realising next-generation networks (NGN)s, such as networking technologies perceived as primarily constituting networks of the future (Assi et al., 2001) and network transport protocols perceived as critical to Quality of Service (QoS) (Flegkas et al., 2002). In addition to varying strongly in the underlying technologies, management frameworks also differed in the scope of their management solutions. Some focused on the need to create end-to-end QoS connections through the network (Engel et al., 2003), while others incorporated broader aspects of customer interaction and management (Cortese et al., 2003). Similarly some groups advocated the need to manage right out to the equipment used at the customer premises

(Vandermeulen et al., 2002), while others concentrated on the need to manage purely carrier infrastructure from the local exchanges inwards towards the core networks (Triminitzios et al., 2001).

While appreciating the unique focus of each architecture, the group viewed them as still addressing a common problem of NGN management. All of the architectures could potentially contribute to the solution though had to be understood for the specific problem they addressed and their driving quality attributes. When starting anew, the predicament becomes how to make use of all of these designs given they may address different subsets of the one problem and overlap in numerous ways. The technology focused solutions tended to focus on qualities like performance and availability (Engel et al., 2003), whereas the broader management frameworks often placed importance on qualities like openness and interoperability (Vandermeulen et al., 2002). The focus on these particular aspects of quality allowed the group to discern the strategies used by these architectures to achieve their quality aims, however it soon became evident that there was generally little attention given to the trade-offs these might incur. Having some degree of insight into the design challenge ahead, the group began to raise concern that the architectures did not address all of the issues associated with their respective design approaches. For instance a focus on performance could impact the ability for the network to personalise service offerings, similarly decisions to increase the availability of the management framework could degrade the timely performance of the system. The way in which these other qualities were maintained within the solution or the rationale behind incurring their degradation in favour of other attributes was seldom discussed.

The trade-offs revealed in examining the existing literature also held significant consequence for the types of services that could be delivered across next-generation networks (NGN)s. For example 'on demand streaming services' such as live video were seen as highly reliant upon the qualities like timely performance. However it was seldom discussed in the NGN management literature exactly what services were envisaged as forming part of their NGN service sets. The quality needs of an NGN management were clearly contingent upon the services and platforms that comprised a NGN, yet as the industry partner stated to us there was no certainty as to what these would be.

The prevailing uncertainty about what constituted the characteristics of an NGN as well as the need to build a more comprehensive picture of the trade-offs in existing design approaches resulted in two distinct exercises for the design group. The first was a visioning exercise aimed at describing the characteristics of the problem situation in a way that would help guide the future derivation of requirements. The second task was aimed at exploring the trade-offs inherent in design an NGN management framework. The best way of achieving which was seen as embodied in the activities of the architecture trade-off analysis method (ATAM).

4.5.Applying ATAM as a design learning technique

The term design learning is used here to generically describe the types of activity conducted early in the design lifecycle. Activity that includes aspects such as: developing an appreciative understanding of the problem at hand (formulation); drawing similarities with other design situations (analogy (Joseph, 1996), idioms (Sargent, 1994)); coming to understand the uniqueness of the design situation compared with the broader analogy(naming (Schön, 1991)); examining approaches to these analogies in order to understand their context as well the characteristic problems they address (framing (Schön, 1991));and conjecturing solution concepts (prestructures (Roozenburg & Cross, 1991)) which need to be analysed and evaluated. Reflecting on the formulation of the situation at hand the designer will have developed knowledge of the way in which the problem can be described and the consequences of subsequent solution approaches.

Early design activity in the collaborative project exhibited many of these aspects of design learning. Design meetings often explored the formulation of the problem in discussions with the industry partner. Analogies were gathered from other aspects of telecommunications and broader disciplines such as market economics. Eventually with a concept of the problem frame, solutions addressing aspects of the problem were gathered in the form of published architectures and discussed within the group. Progressing from the problem formulation and collection of solution fragments, the group sought a means through which they could analyse and test the understanding generated by this activity. ATAM offered itself as a useful technique in achieving these aims, with its characteristics well suited to assisting design learning activity.

The ATAM process elicits and explores both the system need and the solution architecture as well as providing for their analysis with respect to one another. Akin to drawing together the formulated problem and the solution concepts as well as providing a platform for their analysis, ATAM offered to draw a broader coherence over the initial design activity and help document the learning outcomes. In this case the outcomes would include a view of what other industry and academic research groups perceived were the driving quality aims of NGN management, how they were best fulfilled, what were the trade-offs likely to be encountered, as well as a clearer insight into the industry partner's needs and expectations.

From the set of available architectures gathered by the research group, five suitable solutions were selected for the analysis process. The selection was based on criteria such as the amount and detail of public information on the architecture, as well as the circles within which publication was achieved. A bias was given towards material found in well respected industry and academic forums. Granted the requirement within our own design brief to investigate the application of policy-based management, each of the architectures also had to have made use of the Internet Engineering Task Force's IETF's draft policy framework (Stevens & Weiss, 1999) in their design. The architectures were divided amongst members of the design group who were tasked with taking on the role of surrogate architects. They were asked to gather as much literature as possible on the candidate architecture and present it back to the group using some guidelines developed by the lead architect. Based on the recommendations for step 3 of the ATAM (present the architecture), the guidelines included aspects such as the driving quality attributes, architectural strategies, functional descriptions and usage scenarios overlayed onto the system structure.

The ATAM process decided upon was derived from the specification but adapted to allow for the fact that the developers of the architecture would not be present. Typically in the ATAM process, step 3 presents the architecture to the group and some informal analysis of it takes place in step 4 utilising techniques like architectural styles. A more formal analysis takes place in step 6 after the utility tree has been constructed in step 5. In attempting to develop functional examples of the quality requirements and identify what architectural decisions were most responsible for their satisfaction, the step 6 analysis demands knowledge of the architecture that the design group anticipated it was unlikely to develop. Therefore only the first 5 steps of the ATAM process were applied as depicted in Figure 4.5.

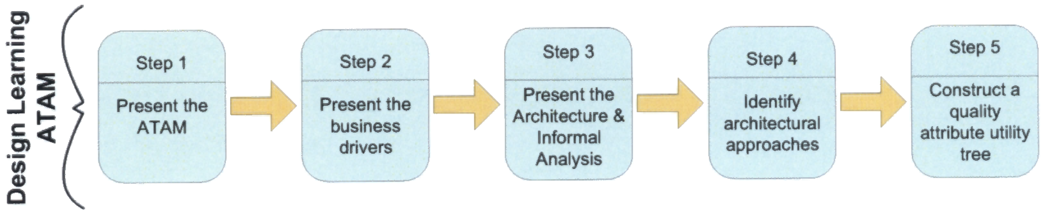

Figure 4.5 - ATAM process undertaken by the design group

Traditionally the ATAM is split into two overlapping phases, due to the fact the stakeholder group wasn't going to change from one phase to the next, only the one pass of each step was conducted. Time constraints precluded all the steps being completed contiguously in one session or even day, therefore the method was split into 3 sessions.

The first session contained all of the 'presentation' activities (Kazman et al., 2000) including the presentation of the method itself to the wider group, the presentation of the business strategic information and the presentation of the architectures.

The second identifies styles within the architectures and establishes their contribution to system quality attributes.

The third session is focused on the 'investigation and analysis' (Kazman et al., 2000) activities. Analysis activities include generating the key system quality drivers and scenarios to complete the quality attribute utility tree.

Having assigned the architectures to the design team members and given them time to prepare reports and presentations the first ATAM session was held. Following an introduction of the ATAM process by the chief architect, the industry partner presented on the business strategic viewpoint, which concluded with an open discussion of any issues stakeholders wished to raise. Each architecture was then presented in turn and discussed by the group with a question and answers style approach. After a lunch break dividing the first and second sessions the design team discussed the virtues of the various architectural styles as they had been applied in the management system architectures. Prior to the discussion the lead architect presented on the notion of styles themselves. Introduced in Chapter 2, architecture styles are commonly occurring patterns of components and rules for their interaction that provide specific quality benefits and drawbacks (Klein & Kazman, 1999). A reasonably comprehensive set of styles and their respective properties were given to the group in order to facilitate discussion.

Due to time constraints the third session aimed at building the quality attribute utility tree was held a week after the first 2 sessions. Each participant was given a sheet on which they were asked to nominate 10 or less of what they perceived as the most critical quality attributes for the system. In a slight deviation from the ATAM method they were also asked to write a brief explanation of the meaning of the attribute as well as provide a priority rating for it. The result was a large number of reasonably diverse attributes.

To cope with this multiplicity the lead architect suggested the attributes should be rationalised and presented back to the group. The rationalisation involved tallying the quality responses in front of the group then shaping them using the semantic of the definitions provided as well as some standards-based knowledge of quality requirements such as the guide for developing system requirements specifications (IEEE, 1998a). What eventuated were 6 core categories (types), whose meaning was seen to adequately function as a container class for the various quality attributes suggested by the group. For instance the category of 'change' was seen as embodying the capability of the system to change and as such incorporated qualities such as portability, maintainability, etc. The quality categories were then related to one another and the overall picture presented to the design team.

As is requisite for the ATAM process, at the same time as eliciting quality attributes, scenarios were also gathered from the group. However the format for the scenario gathering was not one of decomposing quality attributes using attribute characterisations as proposed in the ATM specification, it was more of a brainstorming session. Each participant was given a template containing sections for 6 key stakeholder groups, they were asked to nominate key use cases for each stakeholder perspective. Difficulties in generating the scenarios prompted a second scenario exercise. The completion of this final session marked then end of the first use of ATAM within the project. The following section reflects on some significant aspects of what took place. Following this, 'revisiting the framework' presents the learning outcomes obtained by relating these reflections to the theoretical framework of ideas.

4.6. Reflections

4.6.1. Presentation Activities

In order to develop views of the candidate architectures for the analysis, the researchers were provided with guidance on how to develop representations using common architecture-based notation and methods (IEEE, 2000). The notations and methods were targeted more towards diagramming and conceptual guidance than the more formal, mathematical-based notations such as architecture-description languages (ADLs) (Medvidovic & Taylor, 2000). The guidance referred to what sorts of 'views' could be used to represent a system (Kruchten, 1995), what types of 'entities' would be necessary to describe the views and the way in which elements comprising the views should be depicted. The purpose of producing the common views of each of the systems was to facilitate the identification of styles and architectural tactics. Some comparison could then be made between the respective architectures providing some insight on what approaches would be the most beneficial.

Despite the efforts to produce similar sets of views and information on each of the architectures, direct comparison of their traits proved quite difficult for the group. As discussed in section 4.4 the architectures had to be considered with respect to the motivating problem behind their development. Viewing the motivating problem as a means of differentiating between the architectures meant that systems quality considerations had to be reconciled against the problem solved by the architecture. Rather than simply considering if a certain quality attribute was desirable, the group was challenged to consider whether the perspective on the problem was one they shared. In considering the importance of the motivating problem to the qualities of the architecture, the group began to discuss the need to understand the business perspective as part of the problem.

The scope of the NGN problem adopted by the architecture as well as their key system qualities seemed to hold implications for the business model of a telecommunications carrier implementing the system. For instance one of the architectures took a layered approach to providing NGN services, splitting the service provision into three layers concerned with network connectivity, network resource provisioning and service usage (Vandermeulen et al., 2002). The separation of these concerns promoted a business model where different aspects of the service could be provided by different companies. The need to integrate the different layers saw a focus on qualities like interoperability. However the overhead of integrating between service layers and across companies would make the task of dynamically provisioning services extremely difficult. The architecture implied a multi-provider business model offering pre-defined and pre-configured services. The flip side of this model was evident in another of the architecture which focused on the need for dynamic management (Engel et al., 2003). The lack of consideration for issues of openness and interoperability encouraged a single provider model of service provision. Understandably the qualities for such a system centred on the need for performance. At a systems quality level, what appeared as a trade-off between interoperability and performance, could be seen to hold consequences for an underlying business model.

Through discussion at this level the group came to view the way in which the literature reported the architecture & the qualities it prioritised as being influenced by an underlying business viewpoint. This further augmented the analysis concerns of scope and quality with yet another aspect. Understandably the analysis had trouble focusing in the systems quality domain traditionally dealt with by the ATAM because of the constant need to ground considerations in the motivating problem. Instead of understanding the architectures in terms of well known requirements such as performance or availability, the analysis constantly drew back to business concepts.

Understanding the technical nature of the architecture and addressing trade-offs such as timeliness (performance) versus level of encryption (security) seemed insufficient. Discussion gravitated back to issues like what the architecture allowed the carriers to achieve in a business sense or what they allowed the users to achieve, or what the operational consequences may be for systems, training, hiring, etc. For the NGN design problem the systems quality considerations seemed inherently tied up with the complexity of business viewpoint.

4.6.2. Addressing Architectural Styles

Following the presentation of the architectures the group began the task of attempting to identify architectural styles within the systems presented. To assist in the process a set of common styles were defined in a presentation by the lead architect. From the outset the group had significant difficulty clearly identifying styles in the architectures. While sometimes there was explicit reference to the use of a particular style in the system literature, styles were largely implicit in their design. The task was left to the group to infer styles from the structure of the components in the architecture and the descriptions of how they interacted. Compounding the difficulty was the fact that many of the styles were derivatives of a more abstract configuration of components (Fielding, 2000)). For instance the client-server style could be specialised into layered-client-server, stateless-client-server, etc. Therefore the omission of a small amount of architectural detail could prevent the proper identification of a style. Importantly for the analysis these style specialisations offer differing quality characteristics to the parent style.

Assessing the overall impact of a certain style on the quality attributes of the system proved as problematic as identifying the styles themselves. In some instances the configuration of two discrete components could be classified as complying with a certain style, however it is rare that system operational flow is confined to such small numbers of components. Likewise It is possible to apply different styles in a subsystem than at the system level (Bosch & Molin, 1999), and indeed was found to be the case here. As such it is difficult to understand how the interactions of multiple component relationships and styles contribute to a greater set of properties. To illustrate this example consider the 'client-server' style commonly discussed in literature. A well known example of the client-server style is the connection of a client application to a server application such as a telnet session. The system view of such an interaction might consist of the user, the computer they are using to activate the client, the network connection to the server and the server machine and application. The operational flow is quite contained and simplistic (a user initiating a session), in this way the consequences of the client server pattern on the system are significant and can be clearly understood. Yet such simple operational flows were unrealisable in the NGN system. For example a policy client, may be connected to a policy server, the client of which is connected via an event-based pattern to a local service gateway, which in turn is connected via client-stateless-server pattern to a user premises device, likewise the policy server utilises a blackboard-based pattern to interact with a repository and is peered in a brokered distributed-objects pattern. All of this is only a sub-set of the operational processes to handle the seemingly simple situation where a user turns on their IP television.

Equally as inhibiting is the fact that although architectural styles place constraints on the configuration of components and in some regards their patterns of interaction, there are still aspects of the style that are open in terms of their implementation. These decisions complete the picture of the behavioural aspects of the style and in terms of architecture analysis represent questions that need to be further probed by the analysis team (Kazman et al., 2000). In the context of policy-based NGN management a common issue was the performance aspects of the policy distribution and execution. In most instances the policies represent the behavioural guidelines for network elements in how to implement a service for a user. Architectural styles will stipulate the way in which the policy server and the policy enforcement points are connected, yet details such as 'the amount of policy logic on the target elements', 'the complexity of the policy language', 'the complexity of the information model storing the models used to create policy', 'the algorithms used to compile and check the policies', all have significant impacts of the behaviour and performance of the system. Naturally in the group's situation, clarity was unable to be sought from the design teams that produced these architectures. Consequently the analysis of the architectures spoke of risks and concerns, as well as the disposition of the design towards a specific design aim. The experience suggested where the system is sufficiently complex and irreducible to interactions between a few key components, the abstract nature of architectural design is likely to inhibit the assessment of a system on the basis of styles.

4.6.3. Investigation and Analysis

The investigation and analysis activities of the ATAM focus on the construction of an attribute utility tree. The role of the utility tree is to clearly articulate the quality requirements of the system and derive scenarios for them (Kazman et al., 2000). Eliciting the quality attributes and scenarios provided some important feedback on the ATAM process. Although it is not specified as a requirement of the ATAM, collecting definitions for the quality attributes proved to inform the entire exercise and shed light on semantic issues with the generated artefacts.

A prominent feature of the quality attribute definitions provided by the participants was the apparent difference in the terms used to describe the business drivers compared to the quality attributes of the system. In order to construct the utility tree, the ATAM requires a fulfilment relationship to be drawn between the business drivers and quality attributes. The ATAM literature offers a fairly close relationship between these two informational elements, in many instances proposing what are more commonly recognised as quality attributes, as business drivers.

> *"For example, in an e-commerce system two of the business drivers might be stated as: "security is central to the success of the system..."; and "modifiability is central to the success of system....""* (Kazman et al., 2000, pp 16)

Contrary to these examples the needs synopsis presented by the industrial partner proved difficult to understand in system quality terms. The business drivers emphasised the needs of cost management, including both operational and capital measures, as well as customer choice and satisfaction. These drivers are all contingent on the way in which the system is designed but are nonetheless, quite disparate from the quality attributes nominated by the participants as being critical for the systems success. Characterisations could be attempted in order to draw relationships between the business and quality viewpoints. However through discussion it became clear that assumptions may generally have to be made in order to do so. For example in the NGN solution space, systems commonly refer to the need for high levels of throughput and performance to ensure customer satisfaction (Sheridan-Smith et al., 2004). Yet there are no guarantees that a performing system will be the determining factor in the customer's view of the service. The customer may be happier with a low performing cheaper service, or a service delivery method that does not have any real-time implications. The concept of Quality of Experience (QoE) acknowledges that the customer does not just use technology but lives with it (Keinonen, 2001). Consequently quality aspects associated with the usage perspective partially influence, but aren't solely responsible for the customer perception (Sheridan-Smith et al., 2004).

Similarly from a network carrier perspective the industry partner indicated that carriers would be interested in re-using existing infrastructure if effective. Such a requirement appears to align with the typical quality notion of interoperability. Yet a system that doesn't re-use any infrastructure, but instead makes greater revenue to compensate would be equally acceptable. In both these cases there is an element of wanting to know more about the system in terms of cost and functionality before committing to the proposed solution. It also suggests that care needs to be taken in testing the assumptions behind framing the problem in a particular way. Viewing the customer satisfaction as largely a network performance issue narrows out of view other contributing factors to customer experience like ubiquity, cost, cultural appeal, importance, etc (Armbruster, 1997).

As well as potentially narrowing the problem frame, adopting a particular quality viewpoint has the potential to stifle more diverse quality concerns. Evidence for a more commercially focused business viewpoint, apart from the systems quality viewpoint was also evident in the quality attribute responses of the industrial partner. The language and concepts expressed in their quality attributes did not concur with the more popular technical definitions of systems quality. Although the architecture literature acknowledges the potential for parochial attributes (Boehm, 1988), it is difficult to find literature that discusses or deals with quality attributes beyond the scope of (McCall et al., 1977) and Boehm's (Boehm, 1989) quality models. Attributes within these models include but are not limited to maintainability, portability, reliability, efficiency, etc. In contrast the industrial partner offered qualities such as 'customer/user aware', 'self-assured', 'coherent'. These qualities represented concepts like ensuring the 'coherent presentation of the user as managed within the network'. In order to utilise the responses in the context of the other quality attributes, the lead architect attempted to talk the responsible participant through their responses in order to unveil aspects of similarity with the exiting qualities. Although the lead architect was acting with the intention of being inclusive, the end result did not seem to achieve it. The participant was faced with either reformulating their responses or have them exist as a singularly mentioned (therefore not very well supported) quality attribute, that most likely will be lost in the rush to handle the most well represented qualities. As such their responses were smoothed over and assimilated into the general listing, which tended to lose much of the meaning behind the original participant contribution.

The placement of ATAM within architecture practice tends to presuppose the use of common architectural quality concepts. In this instance these quality concepts showed to be impermeable to the every day business concepts of the industrial partner.

In addition to the differences between the systems quality and business concepts, the responses also showed variability in the semantic constructs used to define the qualities. The definitions brought to light the different perspective of each participant as well as their diverse personal experience. Consequently the quality attribute definitions tended to vary, and vary in important ways that could potentially create ambiguity for the ATAM process.

A notable difference was the different language used to define the qualities. In some instances, definitions were couched in highly 'architectural' terms using well known architectural constructs such as 'connections' and 'components' (IEEE, 2000; Perry & Wolf, 1992), whereas others used more telecommunications system oriented terms such as 'protocols', 'managed data', '5 9's' and 'Frame Loss Rate' (ITU-T). These constructs reflected the diversity of the group, with many of the participants having either traditional telecommunications backgrounds or experience in architecture-based design of computer-based systems. As discussed above, the business stakeholder's attributes also strongly represented the difference of their perspective from the more traditional systems quality viewpoint.

Another salient characteristic of the responses was the use of different quality terms to describe the same phenomena, and conversely the use of the same quality attribute to describe different phenomena. For example several respondents specified bounding the delay, loss and jitter experienced by users as important system requirements. Some participants labelled this need as 'performance', whereas others utilised the term 'availability'. Conversely some participants utilised the same word such as 'scalable', however specified different aspects of scalability. For instance some responses spoke of the need to scale to accommodate more users. Others spoke of the need to scale to incorporate more system components or functionality.

When the quality attributes were being collated in front of the group, several of the suggestions prompted lengthy discussion amongst the participants. These particular quality responses seemed to capture an angle of the system that had not been widely discussed. The level of interest and dialogue that took place showed the participants hadn't previously considered the attribute, yet were drawn in by the concepts it brought to light. Unfortunately in voting situations, participants cannot revise their initial thoughts on the basis of others input and ideas. As Joseph suggests, commonly in design concepts need to 'challenge the brief', to think outside the well defined boundaries of the problem for a novel approach and way of viewing the situation (Joseph, 1996). Before decisions of priority are made in such democratic design situations, there needs to be adequate sharing of ideas to open up the problem situation and let participants appreciate the merits of other's ideas.

Following the development of the key quality attributes, the next task was to try and formulate some use-case scenarios and associate them with a quality aspect of the system. The first step was to generate the scenarios themselves which was done by providing the participants with a set of stakeholder perspectives and asking them to develop a usage scenario from the perspective of that stakeholder. The scenarios submitted by the group members tended to reflect the well established operational and functional descriptions of telecommunications management, such as the FCAPS[9] (ITU-T) model and the Telecommunications Management Forum (TMF) operations map (TOM) (TMF, 2000). Although similar in nature to usage scenarios, these descriptions were closer to abstract functional descriptions that generalised the actions of the user. An example from the user perspective was the common response of 'user adds or modifies a service'. No participants were able to specify what the user interacted with to perform the operation, what the service may be or what the given response could be. Explanation for this can be offered in that the description of the system that would grant clarity to these situations did not exist. No real precedent for such a telecommunications system existed, therefore notions of use which would develop from detailed system knowledge were unclear. In their place abstractions of use, in this case the operational aspects were put forward. The problem with operational specifications like those so commonly used in telecommunications is that they are solution agnostic. They specify what has to be done but don't give clarity on how it should be achieved. A task like the ATAM really requires the detail behind how things

[9] FCAPS stands for Fault, Configuration, Accounting, Performance, Security. These represent key management operations of a telecommunications carrier

are achieved to understand the quality ramifications. Two systems could quite readily exhibit the same operational characteristics but have two entirely different systems implementing them.

The abstract operational nature of the interactions made them difficult to intuitively associate with the quality attributes. A second exercise was conducted where the group were provided with the scenarios elicited in the first exercise. They were asked to nominate up to four quality attributes important for each scenario, as well as specify their reasoning in doing so. An important observation of the responses is that they were sometimes inadequate to really describe the purpose for which they were selected and more concerning was the selection of qualities based on inner systems function rather than externally observable characteristics. A good example was the common quality response of 'change' justified through reasoning such as 'able to meet changing demands' for the scenario 'user modifies their profile' (changes subscription packages and conditions). The response of change implies that the participant was led by the fact a change in the system was occurring, whereas in this scenario the user is simply 'using' a commonly discussed capability of the system, the system itself was not changing. As such the focus of the quality concerns would seemingly be more suited to the usage aspects of quality such as 'usability' and 'responsiveness' rather than change.

4.7.Revisiting the framework

As discussed in section 4.1, learning in an action research project can be made transparent by placing it carefully in the context of a theoretical framework. The link between the theoretical basis of the research and the framework grounds the learning in a greater body of knowledge that is informed by the experience and in turn informs its interpretation and future action. Several key themes were distilled from the theoretical elements of the research and illustrated in Figure 4.2. The following section will discuss these framework elements with respect to the reflections.

A prominent theme to surface within the reflections is associated with the notions of Weltanshauungen and object worlds. As discussed in Chapter 2, these concepts embody the idea that problem situation can be looked at from different viewpoints, revealing a different perspective on the situation (Checkland & Holwell, 1998; Jackson, 1988). Where these perspectives show strong diversity there is an increase in situational complexity.

The tasks of developing the business strategic viewpoint, followed by the task to build a systems quality view helped illuminate the different ways of understanding the NGN management system. These were embodied in the project design group by the stronger commercial focus of the industrial partner compared with the systems quality view encouraged by the academic group's experience in software architecture. Although the software architecture literature suggests one is directly derivable from the other, experience showed that the reconciliation between the viewpoints was not easily achieved. The business language used concepts of cost, customer satisfaction, and simplicity which aren't all easily defined, let alone in terms of a system's properties. Conversely, the quality perspective covers those attributes well defined in systems literature but seldom related to business needs.

To link these perspectives for the purposes of analysis, the ATAM literature characterises business aims in terms of quality concerns. However section 4.6.3 provided evidence of how assumptions are inherent in such a characterisation. In being asked to decide between qualities the industry partner was being asked to answer questions like 'would a customer prefer a cheaper less personalised service, or a more expensive highly tailored one?'. The answers to questions like these are highly unclear in design situations like the next-generation network management system, because not strong precedent existed for the system. The consequences of decision making at this level are further complicated by the realisation in section 4.6.1 of how trade-offs at a system level could affect the business structure of the telecommunications carrier. The capability of telecommunications businesses is very closely linked to the capability of the management system (Hope & Nichols, 2002). Exploring the solution space helped the industry partner discover the bounds of system capability, which they then use to reflect on their business viewpoint. In order to seek better understanding in these situations, answers are sought to more detailed questions such as 'how much more would the service cost?' or 'how much better performing will the system be?'.

Seeking to understand more of what a system can do so that it informs the client's aspirations as to what they are trying to strategically achieve in a business sense unveils the concept of design as exploration (Sargent, 1994). Such exploration inevitably reveals non-obvious results, in likeness of Schon's unintended consequences of design (Schön, 1991), whereby the learning that occurs can influence the way in which the system is viewed. In the event that the business has a clear strategy and view of how to implement it, quality constraints may appear cogently derivable from the business viewpoint. However in this case, the industry partner was clearly using exploration into the technology and systems space to inform its business concepts. Analysis and further understanding becomes difficult as decisions in one perspective are seen to be dependant on decisions in the other. The indivisibility of the problem and the solution conspire to block attempts to refine the problem, which can force assumptions through implicit decision making. The problem and solution appear to evolve together and become concomitant. In software design the learning loop is perceived as taking place between the requirements and the design artefacts (Fischer et al., 1995). While this is indeed important, experience here has shown that the requirements embody an approach that attempts to resolve a business need for the industry partner. Therefore the loop of learning between the original motivating problem and the approach lies as much between the aspirations of the industry partner and the driving requirements of the system as it does between requirements and design activities. Potentially it is even more critical at this stage as the following paragraphs attest that it bridges perspectives as well as from problem to solution.

Examining the process of designing as undertaken by individuals in the situation, Bucciarelli developed the concept of object worlds to embody the notion that individual designers inhabited different 'worlds' based on their experience, skills, responsibilities and interests (Bucciarelli, 1988). The manifestation of which in designing is that different participants working on the same object of design will 'see the object' differently (Bucciarelli, 1988). Later he extended the object world concept to explore the idea that designers effectively speak different languages, not in a natural language sense associated with ethnicity, but a specialisation of normal language that affixes special meaning and symbolism to terms in an idiomatic way (Bucciarelli, 2002). Due to their non-specific nature, the use of quality attributes in a systems context lend themselves to being interpreted and understood in different terms by different stakeholders. Several of the exercises surfaced the diverse understanding of system quality that existed within the group. The quality attribute definitions showed that the language and concepts used to describe the quality varied significantly from one participant to another, often reflecting their experience and concerns. The assignment of meaning to terms also proved to be somewhat ambiguous where participants used the same quality label yet gave appreciably different definitions and conversely used different quality labels yet gave similar definitions.

In order to resolve the ambiguity that can arise in discussing quality attributes, designers are advised to seek the usage aspect to the system that effectively underlies the quality label. The quality label is viewed as a symbolic term, definable in terms of desired systems behaviour. Descriptions of desired behaviour, commonly referred to as scenarios, are seen to remove the ambiguity from the loaded semantic of the quality label itself (Abowd et al., 1997). Discouragingly both the styles and scenario exercises indicated that operational behaviour, especially in terms of detailed usage scenario can be difficult to envisage where the architecture is seemingly immature. The incomplete nature of the system description presents a challenge to the stakeholders in that they must reconstruct the system from the available information (Galle, 1999). Taking Bucciarelli's original concept of object worlds it is evident that such a reconstruction interprets the system from a specific perspective that is not necessarily shared within the group. The uncertainty at this stage is likely to have affected the scenario and quality exercise where the responses seemed to confuse the properties inner function of the system and the externally observable characteristics to a user.

The unique perspective on the situation offered by the participant's object worlds raises the need to communicate. The dialectic process is crucial for resolving issues early on in complex design situations (Curtis et al., 1988). Whilst the ATAM process facilitates communication in a formal manner there was evidence for the importance of non-structured or unplanned communication. During the quality attribute collation, significant amounts of discussion accompanied the discovery of a quality attribute different from those commonly found in quality models. The negotiation of meaning arising from the discussion is a critical component of negotiation in design and a natural precursor to the negotiation of goals (Bucciarelli, 1988). The difficulty for the facilitator, in this case the lead architect came in attempting to utilise the quality responses that seemed out of character with broader set of qualities. In particular the qualities of the industrial partner, which challenged the generic view of systems quality in favour of more specific qualities associated with the telecommunications business need. In the end they were discussed openly and smoothed into the overall quality model that was developing. Despite being able to cope with a diversity of attributes, ATAM tends to encourage a view of systems quality commensurate with the world of architecture, favouring the symbolism and lexicon of architecture. As a result alternate expressions tend to be viewed as not having legitimacy. The situation appeared to have arisen where the mechanisms designed to grant legitimate representation to a group within an architectural context, constrained them, so they could not contribute without diluting the meaning of what they are trying to convey. In such instances the lack of legitimation can be seen to deprive a given stakeholder group of adequate representation (Jamal & Eyre, 2003).

The actions of the facilitator in this instance also indicate how critical the understanding of the facilitation role can be. Although it is not necessarily recommended as part of their duties, the facilitator took upon the task of mediating the situation in order to try and act in the interests of an important stakeholder. Throughout the ATAM exercise, the help of the facilitator was needed to assist understanding, such as in identifying styles and forming opinions of their impact on the system qualities. Although such understanding is the domain of architecture, and hence the analysis team, the benefit to the other participants such as the business stakeholder and the design team has to be questioned.

Figure 4.6 shows the relationship between the elements of the theoretical framework and the problem situation, based on the learning from the reflections above. The structure is similar to that used by Checkland to in Soft Systems Methodology: a 30-year retrospective (Checkland, 2000).

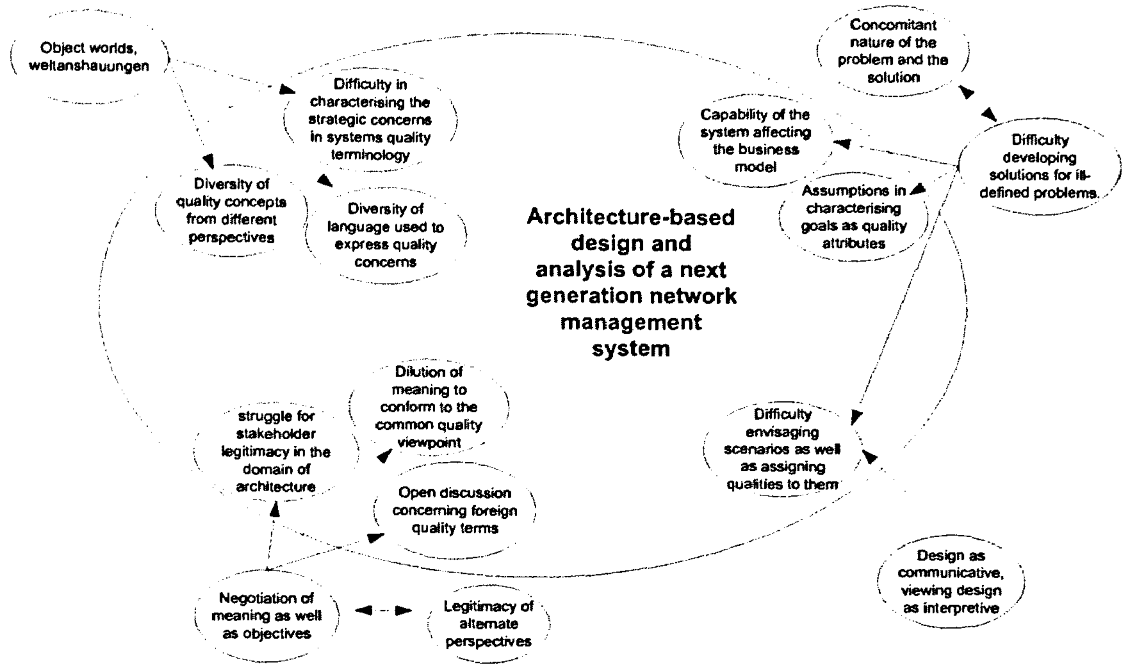

Figure 4.6 - Relationship between the framework and the reflections from the first AR cycle

The framework distinguishes between the theoretical elements of the framework (shown outside the ellipse) and the real world phenomena that drew relevance to these theories arising from the actual problem situation itself (shown inside the ellipse). Several new theoretical elements are evident which have emerged in extending the theoretical scope of the framework in order to explain specific phenomena. The double headed arrows indicate the framework element they extend from. This is not to say these traditional research areas derived from one another, but simply to indicate that in associating the phenomena of the problem situation with a specific framework element, the introduction of this area of theory added something that had not been previously captured.

Summary

Chapter 4 set in place the final elements of the research situation. An explicit theoretical framework was condensed, largely from material presented in the first and second chapters. The rationale behind the use of the ATAM as the architecture-based analysis method of choice was also provided. Prior to engaging in a discussion of the first action research cycle some context was given to the first application of ATAM in the project. This context was described in terms of 'problem-setting', whereby the designers inquire into what is unique about the problem situation and what similarity it bears to other design challenges. Early project work addressed this need by examining existing solution architectures. The result of this examination revealed the concept of a next-generation network (NGN) was not a well defined one and that a large diversity of approaches existed. These approaches were shown to address different quality aspects and enable the achievement of different business level goals. The term design learning was introduced to describe efforts to draw coherence over the vast array of problem and solution fragments that are encountered in early design activity. ATAM was presented as appropriate vehicle for design learning through developing artefacts that reconcile important aspects of the problem and prospective solutions. Creating a platform upon which the industrial partner could discuss with the design team how their ambitions within the telecommunications marketplace would be met by a new management framework.

The application of ATAM in the context of design learning was presented, followed by a reflective look at the exercise. Analysing the reflections with respect to the theoretical framework highlighted how many of the phenomena could be related to aspects of complexity in the ATAM. The diversity of language and concepts offered during the quality attribute exercises drew strong relevance to the notion of 'Weltanschauungen'. With the diversity of perspectives came the need to negotiate meaning, a need made even more acute by the reasonably unfamiliar nature of software architecture. The ability to view systems from different perspectives was also evident in the difference between the business strategic and system quality perspectives. The gap between these two perspectives exemplified the need to move from the customer's view of the problem to the design team's understanding of requirements. However bridging the gap proved difficult with issues at a system quality level commonly reverberated up into business strategic concerns.

The phenomena discussed in the reflections were shown to be well represented by the existing theoretical framework. Consequently this framework is carried forward into Chapter 5. Chapter 5 examines the complexity of the problem perspective through the application of goal-based requirements, which views design trade-offs as characteristic of tension between greater strategic issues.

5

Exploring the strategic perspective through goals

Introduction

Chapter 4 completed the elements of the research situation by establishing both a methodology and theoretical framework of ideas. Candidate methods of architecture-based analysis were then discussed with the Architecture-based Trade-off Analysis Method (ATAM) selected as the most appropriate for the design situation. The ATAM was subsequently used in a design learning capacity to help construct an initial solution space for the network management framework and the learning outcomes presented.

Prevalent within these outcomes was the need to foster understanding of the business strategic and systems quality perspectives, as well as the relationship between them. Chapter 5 establishes this need as characteristic of a broader interdependency in design whereby exploration of the problem and solution domains informs each other. Here this interdependency is presented as occurring between the business strategic and systems quality perspectives to avoid the polemic problem-solution divide. Sound design practice is discussed as balancing the need for exploration of both these perspectives, termed synthesis and analysis. Having used the first ATAM analysis to largely explore the systems quality perspective, the remainder of Chapter 5 discusses the exploration of the business strategic perspective. Goal-based requirements (GBR) are presented as a possible means of establishing the strategic viewpoint and incorporating that knowledge into design activity. The contribution that explicitly modelling goals makes to complex design situations is discussed and potential benefits to the existing design challenge are derived. An appropriate method of GBR is selected from the available literature and applied. Finally the learning outcomes of applying GBR within the project are presented with respect to the existing research framework of knowledge. Figure 5.1 depicts the structure of Chapter 5.

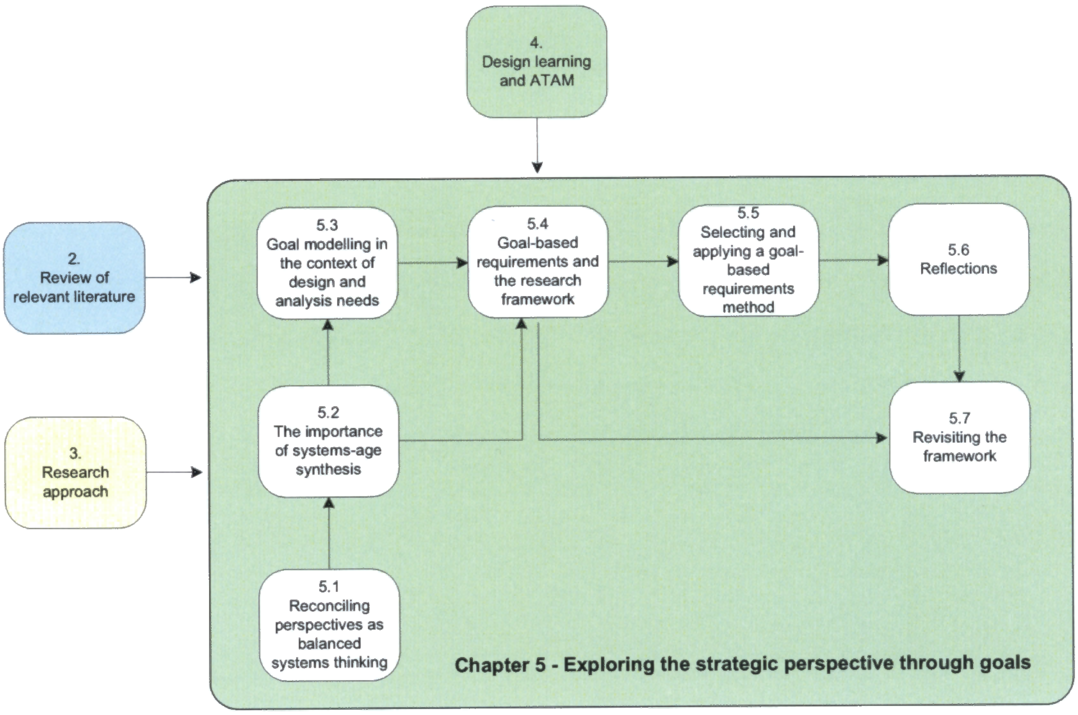

Figure 5.1 - Structure of Chapter 5

5.1. Reconciling perspectives as balanced systems thinking

In reflecting on the use of the Architecture Trade-off Analysis Method (ATAM) as a vehicle for exploring the initial solution space, Chapter 4 surfaced incompleteness in the concepts driving design effort. System quality attributes could not be considered in isolation from the motivating problem, a unique version of which was evident in each candidate system. The architecture also appeared to be affected by, and hold consequences for, an underlying business perspective that hadn't been properly expounded. Attempts at understanding this interdependency revealed a difference in nature between the business strategic and the systems quality viewpoints. A difference reflected in the group composition through the perspectives (object worlds) of the industrial partner representatives and the academic design team members. Consequently during the systems analysis, discussion at a systems quality level tended to be drawn back to issues at a business strategic level. Being able to convey learning from one perspective to the other appeared fundamental to building understanding within the group and facilitating communication between the members of these object worlds. Communication that is critical in a task like ATAM that seeks to align the system capability with the objectives of the project owner.

The need for communicating between these perspectives is indicative of a broader interdependency in design. Brooks noted some time ago that in planning software design activity allowance must be made for the *"extensive iteration between the client and the designer as part of system definition"* (Brooks, 1987, pp 11). It is commonly infeasible to apply the rational problem solving paradigm, which asserts that the problem definition is supposed to be stable and design is a systematic search of the solution space (Checkland, 2000; Dorst & Dijkhuis, 1995). New requirements will emerge during development that could not be identified before solution fragments existed (Fischer et al., 1995). The problem and the solution ideas need to be refined together, allowing the notional problem space and solution space to co-evolve (Maher et al., 1996).

Traditionally software engineering sees this co-evolution as occurring between design and requirements (Curtis et al., 1988). However Chapter 4 showed how the likely design implications of selecting a particular quality attribute created a desire for understanding the business strategic implications. Instead of between tasks, the iteration could be seen to take place between the systems quality and business strategic perspectives, where understanding is complicated by the potential of one to yield significant insight into the other. Discussing co-evolution in terms of perspectives helps avoid the problematic nature of separating design and requirements (Davis, 2005) and the confusion between problem and solution in prominent design texts (Joseph, 1996).

The interdependency between perspectives means issues of systems quality are likely to force consideration of aspects at a business strategic level that hadn't been previously attended to. New aspects of the design challenge will be named, the problem can be deemed to have changed, and naturally the solution concepts (including quality concerns) will be altered in order to deal with it. In this way each perspective informs the other in a continuous cycle of exploration, learning and revision. In the words of Schon *"means and ends are framed interdependently. And their inquiry is a transaction with the situation in which knowing and doing an inseparable"* (Schön, 1991, pp 165). The return of the design group to the strategic perspective in order to properly understand strategic dependencies brought to light when examining system quality trade-offs exemplifies this interdependency and resultant dialectic. Using Schon's terminology the first ATAM sought to develop the 'means' through analysing system properties and solution concepts, yet couldn't overcome uncertainty within the 'ends'. Accordingly a balance with the strategic viewpoint was sought to produce further understanding of the desired 'ends'.

5.2. The importance of systems-age synthesis

Developing an understanding of the system itself as well as the broader elements of its context (referred to here as the business strategic perspective) demonstrates the need for a balance between two complementary and necessary approaches to systems thinking. Ackoff refers to these approaches as 'systems-age thinking' and 'machine-age thinking' (Hitchins, 1992). Characterising the group's approach to date, machine-age thinking represents the type of systems thinking that has traditionally underpinned scientific and engineering pursuits. It looks into a situation, seeking to decompose in order to explain the properties and behaviour of the contained parts separately. Alternatively systems-age thinking is outward looking, yields understanding rather than knowledge, and attempts to explain the behaviour of a system in *"terms of its roles and functions within its containing whole"* (Ackoff 1981 in Hitchins, 1992, pp 13).

Looking outward from the system itself and building understanding of its context is vital in organisational settings where computer-based systems have evolved from being loosely coupled to the operations of businesses to being 'critically embedded in daily organisational activities' (Bubenko, 1986). Systems are now viewed as enabling innovative business solutions rather than simply automating well established business processes (Yu, 1997). The implementation of such systems braces all layers of the business from strategic decision making and planning, to operational work procedures, right down to the technology domain (Bubenko, 1986). The dependency of the strategic vision on the systems which realise the operational environment is particularly acute in telecommunications management where the capability of the Operational Support System (OSS) *"defines the possibilities and limitations of what the service provider can do. The board and marketing department can define new business strategies and developments, but the reality is that you can only sell what you can deliver, and the determinant of what can be delivered is the OSS"* (Hope & Nichols, 2002, pp 6).

This dependency resounded strongly through subsequent discussions in design meetings and became a dominant feature of the project documentation produced during this phase (Colquitt, 2003; Sheriden-Smith et al., 2003). It was reasonably well accepted market view that the telecommunications industry was facing declining value in its existing service sets (Main et al., 2002). An imperative lay in front of the industry to develop new service types and methods of delivery that permitted carriers to achieve economies of scale in delivering multiple, value-added services, through common infrastructure and technologies. The realisation of new service types and methods of delivery lay contingent on the ability to develop new management capability (Sheridan-Smith et al., 2004; Sheriden-Smith et al., 2003). A new management framework represented more than just a desire to manage the network better, it indicated a greater desire to evolve telecommunications operational and business behaviours (Uglow, 2002), indicating significant strategic value in the system at hand.

In elevating the importance of the strategic viewpoint, the design group came to appreciate that the business objectives of the situation were significantly more complex than they had anticipated. The ATAM literature tends to paint a reasonably simplistic and direct relationship between the strategic business drivers (goals) and the system quality attributes. The depiction of business drivers as very similar to system qualities creates the impression that the strategic issues are well understood and that the strategic viewpoint can be readily characterised and explained in terms of system quality attributes. However experience from the first ATAM led to a less simplistic view on goals. Figure 5.2 shows some business case artefacts, which help illustrate there was at least one level of refinement to the business goals.

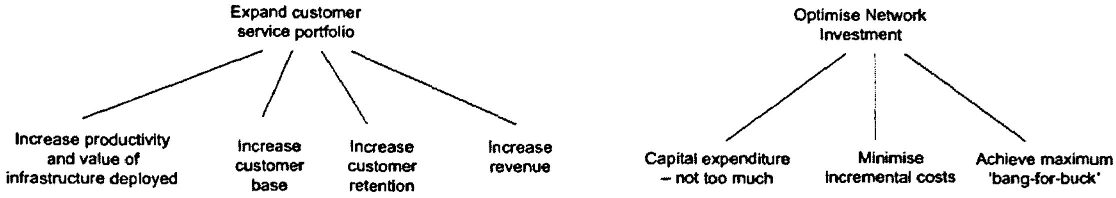

Figure 5.2 - Hierarchy of business drivers from the first ATAM exercise

This experience of strategic complexity was further reinforced by investigations into the strategic issues of the telecommunications market and subsequent design meeting discussions. Competitive and regulatory viewpoints augmented the set of concerns, whilst interdependencies between drivers and the ability to decompose them, increased awareness of how complex they were as artefacts. Consequently, a significant amount of effort was placed into documenting the strategic perspective of the problem situation before it was considered adequately expounded. Two additional project deliverables were produced directly dealing with this perspective. While the exploration of the business strategic viewpoint proved a significant learning experience and raised the group's awareness, ultimately this knowledge needs to be used in design and analysis activities like the ATAMs. The format of the existing artefacts, mostly documents and a few diagrams did not readily facilitate the explicit application of this knowledge in design or analysis contexts. The following section presents a requirements approach called goal-based requirements (GBR), which has been specifically developed in order to ensure the strategic perspective of a problem situation is adequately modelled and translated into design requirements.

5.3. *Goal modelling in the context of design and analysis needs*

Emerging from within the field of traditional requirements engineering (RE), goal-based requirements (GBR) elevate the importance of root justifications in systems design. The impetus for this focus is evident in the earliest definitions of requirements engineering, such as that of Ross who suggests that requirements should capture why a system is needed.

> *"...requirements definition is a careful assessment of the needs that a system is to fulfil. It must say why a system is needed, based on current or foreseen conditions, which may be internal operations or an external market. It must say what system features will serve and satisfy this context. And it must say how the system is to be constructed...."* (Ross & K. E. Schoman, 1979, pp 366)

Despite the overt reference to establishing the 'why' in such seminal papers in the field, subsequent requirements engineering work did not reflect its importance (Lamsweerde, 2000, 2001). RE methods and techniques tended to focus on the need for a precise statement of the system functions, constraints and behaviours perceived as necessary to enable successful implementation. RE methods derived from this view address the 'what/how' aspects raised by Ross & Schoman, leaving a significant gap in the derivation of requirements artefacts that simply explain 'why' the requirements were there (Dardenne et al., 1993; Lamsweerde, 2000; Lee & Hsu, 2002; Yu, 1997). The explicit modelling of goals as part of the requirements process was developed as a way of filling this gap and granting context to the system requirements. Goals were viewed as offering the crucial motivation and rationale behind the existence of the requirements. Their integration into requirements models provided a way of gauging whether the requirements sufficiently address the higher-level objectives which gave rise to the system need (Lamsweerde, 2000).

While goals provide valuable insight into the objectives that are motivating the system need, they are generally ill-defined and quite fuzzy in nature (Holbrook, 1990). Traits like these are clearly not commensurate with the requirements ideals, which include characteristics such as correct, unambiguous and consistent (IEEE, 1998b). GBR requirements frameworks have sought to reconcile the abstract and ill-structured nature of the goals with the concrete and well structured nature of requirements. Techniques of modelling, specifying, elaborating and verifying goals have all been devised to ensure that knowledge from the abstract, goal level can be correctly imbued in the formal system requirements specification (Lamsweerde, 2001). These frameworks extend the benefits of goal modelling beyond providing a rationale behind the system requirements to fostering a closer association between the goals and subsequent, requirements and design activity.

Understanding the relationship between the system and its objectives permits goals to both inform and be informed by the process of design (Yu, 1997). Inform in that the decomposition and refinement of goals yield system requirements that underpin the design approach (Letier & Lamsweerde, 2002). And be informed in that the impact of issues arising from design and analysis activity can be understood in systems and strategic terms (Dardenne et al., 1991; Yu, 1997). By encouraging the design team to look outwards from the problem (synthesis), while providing a means of utilising this knowledge in decomposing and describing system behaviour (analysis), goal-oriented modelling also facilitates a balanced approach to systems thinking. In this way goal-oriented modelling mediates the dialectic between the strategic and system domains, helping to brace what were experienced to be in Chapter 4, two quite different but interdependent perspectives.

Building on the ability to generate understanding in, and transfer knowledge between these perspectives, GBR has also suggested goal-oriented modelling can be used to detect and resolve conflicts occurring in requirements (van Lamsweerde & Letier, 2000). Trade-offs are inherent in design and conflicting requirements can often be abstracted back to conflicts at a goal level (Dardenne et al., 1993; Lamsweerde, 2001). The first ATAM exercise showed empirical backing for this with considerations at a system quality level commonly resulting in discussion and considerations at a business strategic level. Dealing with trade-offs at a goal level also provides the right level of understanding for key business stakeholders (Lamsweerde, 2001).

The idea of making goals accessible to non-technical stakeholders indicates that goals also have a communicative role to play in requirements engineering. Requirements have traditionally exhaustively described technical systems in a semi-formal or formal manner. Conversely goals are extracted from natural language statements (Lamsweerde, 2003) and are modelled and presented in more comprehensible formats (Lamsweerde, 2001). Rather than immediately moving into the system technical viewpoint, goals encourage the development of a deeper understanding of the strategic viewpoint. Therefore the negotiation of meaning that surfaced in Chapter 4 as a key concept in designing, takes place at a level of abstraction the business stakeholders can be comfortable with. The less technical nature of discussion and less rigid language and representation also aids in communication amongst the potentially diverse and numerous stakeholder viewpoints that exist in project groups (Dardenne et al., 1991; Yu, 1997).

5.4. Goal-based requirements and the research framework

The preceding sections established that goal-based requirements meet a driving need of the project to further understand the strategic perspective and utilise that understanding in design and analysis activity. As well as assisting the project, the discussion also revealed several aspects of GBR that are likely to address some of the key ATAM issues raised in the research framework of Chapter 4 Figure 4.6. The comprehensible nature of the goal models, and goals themselves, helps the stakeholder group both contribute and communicate outside the lexicon of existing quality frameworks and system concerns. The ensuing detailed model of the strategic perspective helps to reveal the true complexity of the goals, the refinement of which provides a bridge between the system requirements and goals. The traceability offered by this bridge allows system trade-offs to be understood in a strategic context, thus facilitating the necessary iteration between the business strategic and systems perspectives. Figure 5.3 illustrates a revised theoretical framework, incorporating the relationships to the GBR concepts discussed above. The complexity of the strategic perspective has also been added as a significant real world phenomenon.

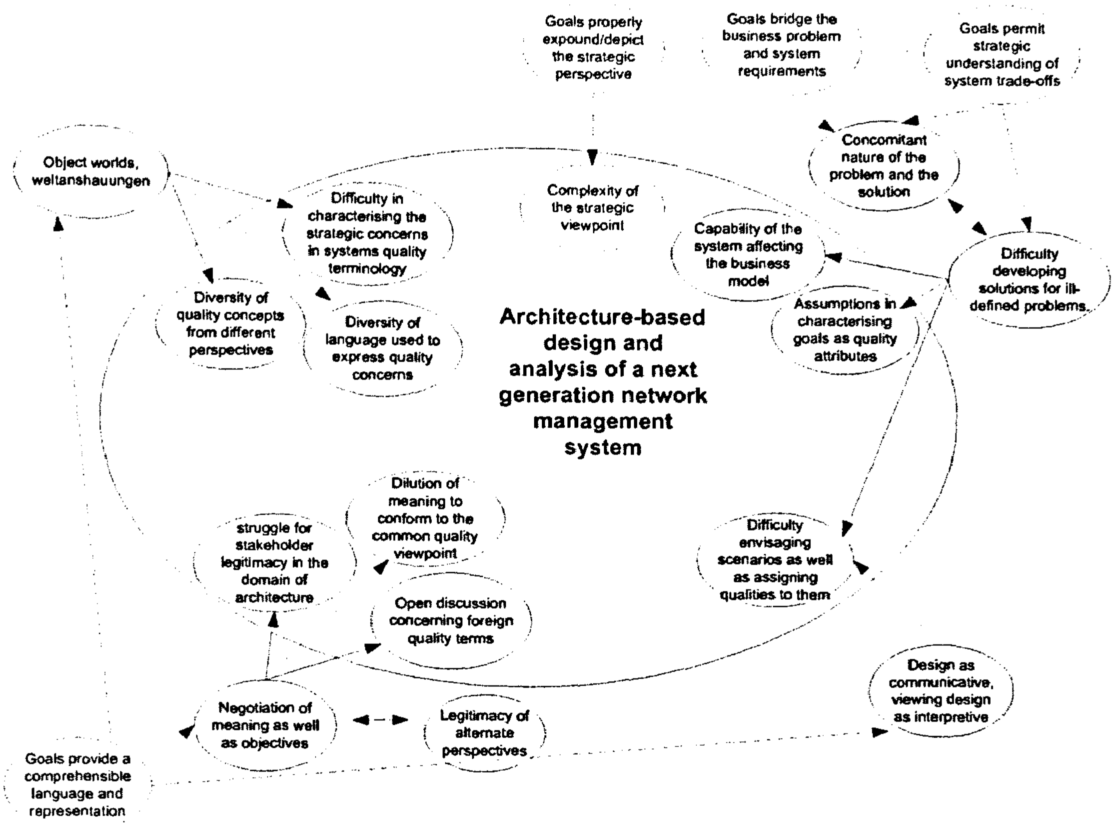

Figure 5.3 - The evolved framework of ideas from Chapter 4 shown with relationships to the goal-oriented requirements concepts

The preceding sections have discussed how the GBR principles will be used to assist the project need to clearly understand the strategic perspective. It has also discussed how goal-based requirements contribute to understanding complexity in architecture-based analysis and demonstrated its relationship to the theoretical framework of ideas. The following section presents the selection of an appropriate GBR method and outlines the stages in which it was applied to the collaborative research project. The remainder of the Chapter 5 reflects upon the outcomes with respect to the framework of Figure 5.3, shedding light on the use of GBR in early design activities and importantly how GBR could be applied to assist analysis.

5.5. Selecting and applying a goal-based requirements (GBR) method

The discipline of goal-based requirements (GBR) has yielded a number of approaches and frameworks including Goal-oriented Requirements Language (GRL) (Liu & Yu, 2004), i* modelling (Yu, 1997), Crews-L'ecritoire (Rolland et al., 1998), structuring requirements using goals (Lee et al., 2001), Non-Functional Requirements (NFR) Framework (Mylopoulos et al., 1992) and the Knowledge Acquisition in autOmated Specification (KAOS) methodology (Dardenne et al., 1993; Lamsweerde, 2001).

The selection of a specific GBR method amongst the candidates was initially based upon whether the method in question demonstrated the GBR capabilities discussed in section 5.3. These include the ability to model high level objectives as goals, the ability to refine goals as well as the ability to model their conflicts and dependencies. Further discriminators in the selection of the method were concerns such as the maturity of a method and its proliferation within associated academic literature. A brief analysis of the methods on the basis of this criteria resulted in the decision to utilise the concepts from the (Knowledge Acquisition in autOmated Specification) KAOS methodology. The KAOS methodology was developed in the early 90's and refined over the subsequent time frame, resulting in continued application and improvement of its process and techniques. Publication achieved within academic literature indicated the methodology was peer respected. Its comparable longevity also suggested a significant amount of both empirical and theoretical work had gone into the development of the methodology.

Although it presents as a methodology, the more recent KAOS literature focuses on the techniques to develop and evolve a goal graph, leaving the process by which they are arrived at implicit in the types of artefacts that need to be produced. However the earlier work by Dardenne, which has stronger process guidance and the later work by Lamsweerde, both seem to agree that an initial goal acquisition phase has to take place (Dardenne et al., 1993; Lamsweerde, 2001). The goal acquisition is followed by refinement and eventually formalisation of the goals. Once the goal refinement has yielded sufficient understanding, goal contribution and conflict relationships can be identified and represented.

Goal acquisition in KAOS is initially focused on developing the strategic perspective by identifying key customer drivers from system concept descriptions where keywords such as 'objective' and 'purpose' have been used. These customer drivers are used to create the most abstract goal artefacts referred to in the KAOS literature as 'soft' goals. The label 'soft' is used to denote a goal whose achievement cannot be unambiguously determined due to a bounded understanding of the situation or contributing factors. A soft goal can only be deemed 'satisficed' (accepted within bounds), rather than achieved (Dardenne et al., 1993; Lamsweerde, 2001; Mylopoulos et al., 1992).

These soft goals form the origins of a hierarchical structure, where goals are seen to elaborate on one another. With each successive refinement more understanding is generated about how the root objectives can be achieved. The eventual aim is to achieve goals of sufficient granularity that they can be assigned to a single system entity to achieve. At this stage, the goal is referred to as formalised and is effectively seen to reveal a system requirement, thus linking the system and strategic perspectives. As exploration of the goals reveals that achieving one goal may hold negative or positive consequences for the achievement of another, goal conflict (trade-offs) and contribution can be derived. The resulting model is a 'goal graph', depicting high-level (soft) goals as 'clouds'. Parallelograms denote 'formalisable goals', while sub-goal refinement is indicated using arrows from one layer to the next. 'Support links', indicating goal interaction are indicated using positive and negative symbols. Figure 5.4 depicts a commonly presented application of the KAOS techniques to model the goals of an advanced train control system, designed to increase the capacity of a rail system in the San Francisco Bay area (Lamsweerde, 2000, 2001).

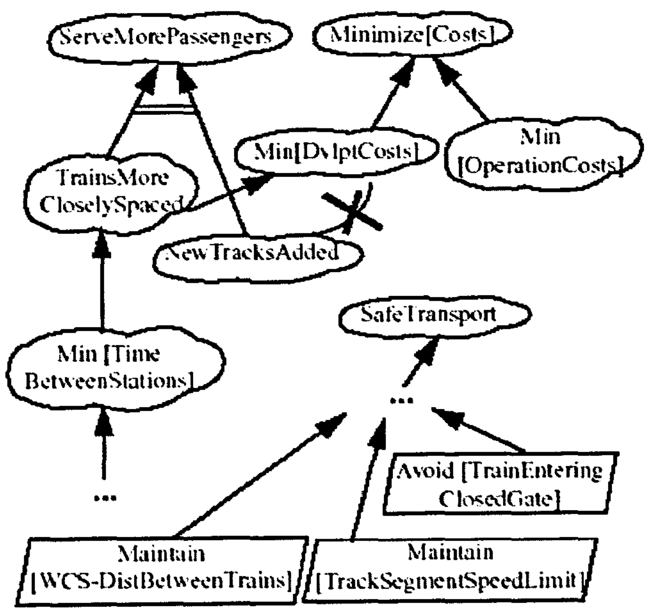

Figure 5.4 - Preliminary (high level) goal graph for a transport system (Lamsweerde, 2000)

Using the concepts in the KAOS, the group embarked on the task of constructing a goal graph in the likeness of Figure 5.4. Granted the complexity and size of the design problem at hand and the number of stakeholders in the group, it was perceived that the goal-based requirements work would have to take place across many workshops. The first watershed in the process is the identification of a series of high-level objectives (soft-goals), complete with positive/negative interactions and refinement hierarchy. Therefore the initial workshop was designed to test the application of the method to structuring the strategic understanding developed for the two strategically focused project deliverables. All that was expected was a small graph of 'soft goals' or 'clouds' as their representation suggests. The workshops were conducted in a brainstorming manner, whereby goals were initially elicited and then later manipulated according to discussion that took place amongst the participants.

As the reflections below recount the first workshop aimed at developing a high-level goal-graph proved more intensive than first expected and the outputs bore a fairly minimal graph with a few business strategic objectives. Several more iterations of the workshop were held in order to embellish the graphs with more detail and work towards a point where technical requirements could be realised (ie goals were 'formalisable'). In a bid to move proceedings along, two of the design team members developed more detailed graphs, which were presented to the group for comment and reformulation. Time pressures and the perceived need to progress the task subsequently saw GBR effort focused on developing these more detailed individual goal-graphs to the detriment of the original group goal graph created in the first workshop. Consequently these goal graphs became key artefacts in furthering the goal based view of the system. Believing the goals were refined enough to attempt formalisation, the group set about deriving some descriptions of behaviour that comprise the constraint element of formalisable goals. Difficulty in formalising the goals led to an alternate method of developing goal strategies that could used as design guidance. The continuing difficulties caused the GBR exercise to be halted with only a high-level goal graph and a series of goal strategies as its output.

5.6.Reflections

5.6.1. Constructing a high level goal graph

The opening GBR workshop started the goal acquisition activities by developing a first pass goal-graph. The preceding weeks of exploring and documenting the strategic aspects of the problem ensured levels of participation were initially quite high. Careful consideration and insightful discussion accompanied the placement and interrelation of each goal. Although dealing with the subjective nature of soft goals, the group was still quite fastidious about exploring and assessing the semantic of each goal as it arose. Even at an abstract level there was a strong desire within the group to ensure that some degree of correctness was discernable from the openly negotiated meaning of each goal.

Established business strategic understanding significantly influenced early goal elicitation, as indicated by the preliminary goal graph created during the first workshop (Figure 5.5). The highest level goals reinforce the importance of issues such as cost and customer experience that had previously been unearthed during the first ATAM exercise. In particular cost concerns were prominent in the root goals of the graph reflecting the strategic value of the management system. Consequently the root goals attempted to address/capture the core interests of a commercial entity, which tended to draw back to profitability. Refining the notion of the profitability reveals a balance between revenue and cost elements (operational expenditure (OPEX) and capital expenditure (CAPEX)).

The perceived importance of the management framework to the revenue aspirations of telecommunications carriers is apparent on the left hand side of Figure 5.5. Here the goals depict an aim to increase revenue by 'enriching' user experience through new service capability. The project title of 'Management of Enriched Experience Networks (MEEN)' conveys the collaborative project vision that these service sets are enabled through advanced management frameworks. The extent to which the goals on this side of the graph were refined shows an initial commitment to this reasoning, compared to the expenditure issues which are shallowly refined on the right hand side of Figure 5.5. Reflecting the view that the system was largely being created to open up new revenue opportunities not being created to address the optimisation of cost aspects (Colquitt, 2003).

Taking an appreciative viewpoint, the goal graph appears to attend more to the functional relations than it does the metabolic ones (Checkland, 2000). The metabolic relations are those designed to preserve the stability of the system whereas the functional ones contribute to the perceived success of the system (Vickers 1974 in Checkland, 2000, pp S50). Design situations like the MEEN project tend to encourage this focus on functional relations, as expectations of a creative and novel solution are inherent in the collaboration between a commercial entity and an academic institution. Issues of a businesses economic viability are seldom in the scope of such endeavour. By refining the goal 'maximise revenue' the group was largely coming to terms with the task presented to them and attempting to generate as much understanding about this facet of the problem as possible. Concerns such as cost optimisation can be seen to shape/influence the system but were not viewed as the principal reason for its derivation. They are the aspects that balance the system, yet those contributing most to the multiple perceptions of success are the services rendered by the system (Regev, 2003).

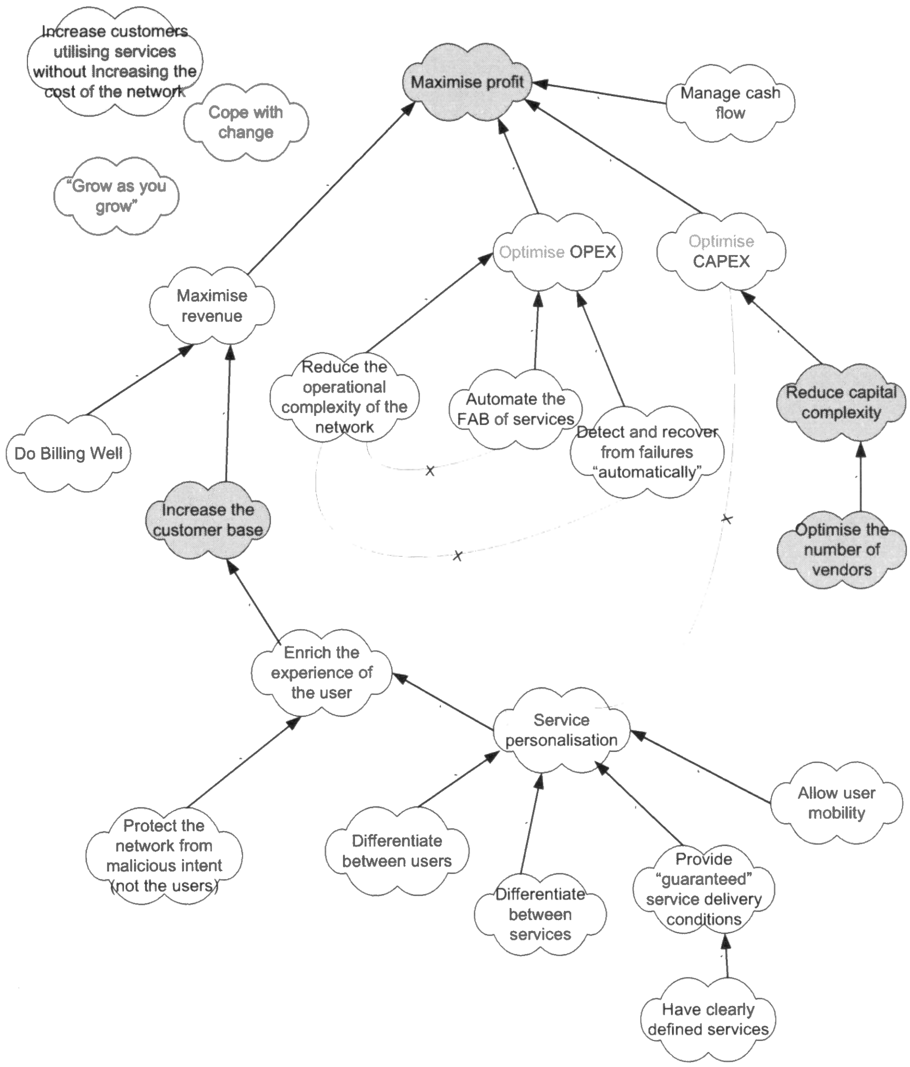

Figure 5.5 - Preliminary goal-graph for the MEEN system

Aside from the cost concerns Figure 5.5 also shows a cluster of goals to the top left hand side that aren't connected within the goal-graph. The goal 'increase customers utilising services without increasing the cost of the network' is captured for posterity but considered to represent a conglomeration of several goals already shown in the graph that cite the aim to 'increase the customer base' and 'optimise expenditure'. However two other goals concerning growth and change were unable to be placed within the existing goal graph hierarchy, yet weren't already represented in the graph.

Essentially these goals expressed the need to change incrementally with the environment of the system, meaning at no stage should it need disproportionate amounts of resources to achieve an incremental increase in capability. The notion was a far reaching one since the resource could encompass high level resources such as operational and capital budget, as well as lower level resources such as the number of servers required to service an increase in voice customers. The goal seemed very similar in nature to a quality requirement without a context, for which there are no known measures for (Kazman et al., 1996). Important as it was, the goal could be related to any facet of the system. The group discussed whether this made it a candidate for a new root goal, though they rejected the idea expressing the belief that it did not appear relevant as a root business driver. The group developed the view this was a general, overarching principle, which was contained enough in definition to be expressed as a goal, yet far reaching enough that it should be kept in mind when developing all aspects of the system.

As the first workshop progressed and difficulties placing and refining goals began to take hold, the dynamic of the group started to change slightly. The large amount of group knowledge on strategic issues and subsequent attention to detail meant that the first workshop did not progress as far as the facilitators had intended. Figure 5.5 shows the preliminary goal-graph created during the first GBR workshop. Despite having derived a significant number of goals for a preliminary graph (23 in total), the level of goal abstraction did not appear to have shifted far from the business strategic perspective. The hope had been that the soft goals could be expressed in a reasonably concise manner and readily refined into more systems level goals, as portrayed in the KAOS literature (Figure 5.4).

The persistence of the goals at a level of abstraction that wasn't satisfactory to realise requirements for a technical system caused the group to voice concerns that the GBR exercise was replicating not adding value to the project strategic documentation (Colquitt, 2003; Sheriden-Smith et al., 2003) . The work undertaken to document the strategic perspective hadn't been forecast in the project plan, and invaluable a learning experience as it was, it was still perceived as adding load onto an already ambitious work schedule. Although the exercise had just begun, the participants were acutely aware that the knowledge presented in the high-level goals, existed to a large extent in the strategic documentation and that the challenge lay in refining them into useful requirements artefacts.

The inability to make significant inroads in to formalising the goals and difficulties experienced with the GBR techniques appeared to trouble the group in general. In particular the design team member facilitating the start of the workshop appeared to withdraw from the process. Whilst the length of the workshop and the difficulties encountered are all relevant to the way in which group members participate, a further contributing factor was later discovered. Prior to the workshop, the design team member had developed a significantly more extensive goal-graph than in Figure 5.5. As the workshop graph evolved away from their previously created vision of the system goals, the team member adopted a quiet demeanour, seemingly unable to reconcile their own goal graph with that of the group. A task that they later undertook, following the workshop they circulated a goal graph merging the workshop output with a personally developed goal graph.

5.6.2. Evolving the goal graph

The desire to negotiate a personal perspective on the goal-based requirements became more evident as the GBR phase progressed. Taking impetus from their colleague's goal graph initiative, and responding to the perceived lack of progress, another team member also set about making a revised goal graph. The team member developed their goal structure from the viewpoint that the goal graph was becoming less comprehensible as it expanded and that the goal refinement process needed to be accelerated in order to yield formalisable goals. The result was large goal graph containing approximately 85 soft goals and highly structured in a tree-like fashion. Subsequent interactions between the members of the design team sparked revisions of both of the personally derived goal graphs prior to the staging of the next GBR workshop.

Having been derived within the same exercise the individual goal graphs bore some significant similarities. Both used the cost-based concerns as root goals and the refinement of these goals tended to cover very similar issues such as diversity of customer choice, service quality, network resource utilisation, etc. The goal graphs were also both split into two main sections, one of the goal graphs explicitly referred to this division as representing a split between operational and strategic concerns, whilst the other referred to the split as occurring between cost optimisation and growth. Parallels with these dichotomies and the structure of the preliminary goal graph in

Figure 5.5 were evident. The operational cost concerns representing the aspects required for stability in a telecommunications business and the strategic-growth concerns exploring its criteria for success.

While these graphs showed the commonality of experience between the different designers they also bore the hallmarks of having been produced from two individual perspectives. The semantic of the goals and breadth of issues incorporated into the goal graphs differed. As did the structural aspects of the graphs such as explicit use of layering and group listing of goals in place of visual tree branching. The pursuit of a distinctly personal representation of the goals by each of the participants suggests that although common levels of understanding can be achieved about a problem situation, the representation of its constituent issues within an artefact is unique to each individual. Developing artefacts in a group workshop provides transparency of reasoning and allows each participant the chance to negotiate their perspective into the end product. The act of producing an artefact helps bring the diverse object world perspectives into coherence (Bucciarelli, 2002). The poor transparency of how these goal graphs were being derived and their difference from the original artefact, increasingly alienated the other members of the group from a task in which they had been originally engaged in. The group's ability to work with the goal graphs appeared significantly diminished.

In a bid to diligently include all the options available for each goal refinement, the individual goal graphs addressed quite a broad range of issues. The group expressed concern that the breadth of issues was attempting to adopt issues that couldn't be realised by the system. These included options to increase revenue such as 'find supportive alliances with vendors', 'discount aggressively', etc. In attempting to comprehensively cover the driving system goals, and refine potential solutions, there appears scope for non system related aspects to appear in goal models. This creates a need to somehow decide what is and isn't relevant as the goal refinement proceeds.

During discussions concerning the goal graph scope, one of the participants also noted how the goal graph didn't appear to naturally represent all of the stakeholder perspectives. They suggested the goal notions as depicted for the business as a whole, did not necessarily agree with the goals that a network administrator may offer. In order to further explore this idea, two of the participants who had a significant interest in the customer view and the network administration view, were asked to develop goals according to those perspectives. The exercise proved quite informative as it provided a timely reminder of the assumptions that can creep into such exercises. The problem of assumptions in refining goals from one level of abstraction to the next held strong parallels with the difficulties characterising goals as system qualities, experienced in the first ATAM (Chapter 4). For instance the goal of 'increase customer satisfaction' was refined into goals such as 'provide service personalisation'. The customer advocate took the view that this was effectively trying to achieve 'diversity of choice'. In associating customer satisfaction and personalisation, the previous graph made the assumption that personalisation was what the customer wanted. Whereas the customer viewpoint suggests they may want diversity, but this need not necessarily require personalisation. Similarly the presentation from the network administration perspective revealed that the network administrator perceived themselves as an entity, which had relationships to the business and the customer. This showed the network administration perspective could not be subsumed under that of the broader business. The problem solving nature of goal-based requirements appears to come to the fore in this exercise. GBR appears to compel the designer to recursively solve each goal, more so than it does to challenge them to entertain different views of the situation.

Despite being significantly more elaborate than the preliminary goal graph of Figure 5.5, these later goal graphs still failed to reach a level of granularity that resembled the requirements artefacts from the KAOS literature. A significant theme from the earlier workshops was concern amongst the group, that multiple goal graphs had been derived, some of which were very detailed and covered many levels of refinement, yet formalisable goals still seemed unattainable. Uncertain of the feasibility of being able to yield formalisable goals in a timely and succinct manner, the design group turned their attention to achieving the goal granularity required to create formal representations. The third workshop was a focused attempt at progressing the goal based requirements work to the point where it could yield contained technical requirements, commensurate with a feasible system design. The process that was derived was to take the leaf goals from the existing graphs (ie the goals that were assumed to be the most refined) and to set them as apex goals for a detailed goal graph. Removing the context to the goals attempted to facilitate a focus on simply refining a few goals to a formalisable level rather than completing the greater graph.

The task proved a difficult one for several reasons, the first arising from the workshop focus on moving from soft goals to formalisable goals. The KAOS literature provided little guidance as to how this step occurred, it seemed to be considered a natural part of refinement, making the workshop difficult to facilitate. By this stage, the researcher had taken over the role of facilitator, problematically they had already begun to develop a personal belief that the KAOS GBR method did not cope well with such complexity early in design. As such their facilitation would indeed be called into question on the basis of this belief. The final point was the removal of the leaf goals from their context within the greater business goal graph. By making them stand alone, apex goals, they lost most of the context granted to them from their derivation within the original goal graphs. A lot of the early questioning from the workshop participants surrounded the parent goals of the new apex goals, trying to resolve perceived ambiguity and perceived overlap between goals.

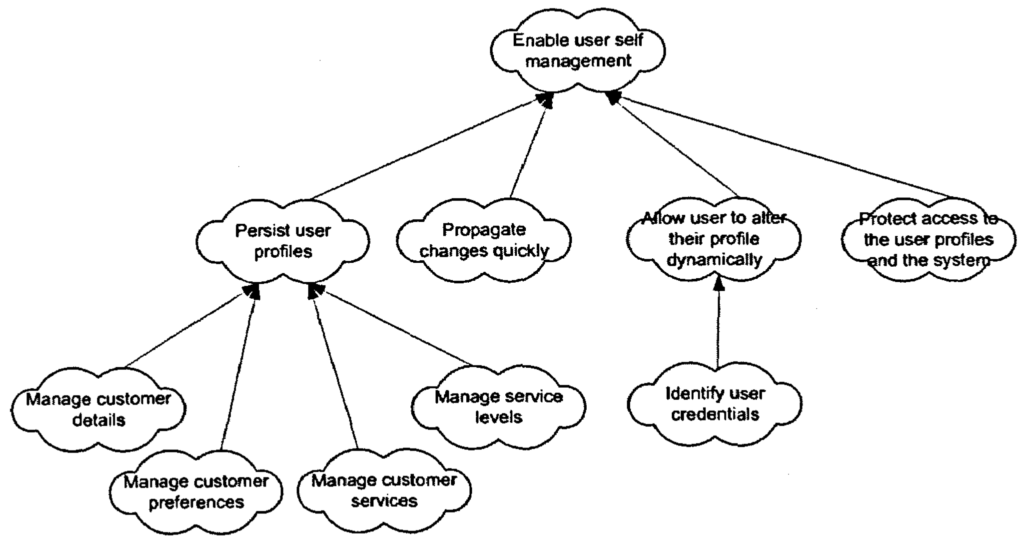

Figure 5.6 - Goal graph for 'dynamically monitor and react to service quality' from 3rd workshop

Figure 5.6 above depicts one of the goal graphs created during the third workshop. The apex goal 'Enable user self-management' had been a leaf goal in the goal graphs derived from the first few workshops. As is evident from Figure 5.6, the refinement of this goal yielded the need to: persist and distribute user information; protect the integrity of the machines performing the actions; and protect the integrity of the information itself. These are all important aims but they don't yield hard and fast operational constraints, from which, one could derive functional and data entities of the system, what the properties of these entities are and what actions they perform.

Quite to the contrary, the group discussed the idea that formalisation at this stage could be detrimental because it would prematurely develop notions of systems objects and functions that needn't necessarily be part of the end solution. For example, the notion of a user profile as the mechanism for persisting and managing user information became an assumed part of the solution. One of the participants presented quite strong views that creating such goals appeared to be making implicit design decisions in the goals. Suggesting that such goals made assumptions about what type of information was being persisted and how it was being used within the system. The notion of information structure and functions to utilise it, were very closely related to the informational aspects of policy, which as the participant pointed out were the focus of lengthy exploration on their part.

The effect of creating tangible objects and implicitly committing to them became clear on examining the goals of Figure 5.6. The group seemed to gravitate towards the notion of a user profile as a combination of service level agreements and personal information. Bundling this information into a profile provides a single mechanism to access and distribute the information. Consequently all the goals were based around how to manage such a concrete thing, ie how to persist it, change it, propagate changes to other nodes, etc. User information would be utilised within the system, but how much and for what purposes, let alone whether it is persisted within the management system itself or indeed exists as a profile entity are all decisions which shape the data and behavioural aspects of the system. Decisions the group was unwilling to make at this stage granted their fairly high-level knowledge of the implications of such decisions.

The resultant uncertainty about appropriate entities and their roles within the system made formalisable goals very difficult to realise. Goals at such a level of refinement required strong notions of the objects within the system, the functions they performed, the actors they interacted with and the data they dealt with. With a goal like 'propagate changes quickly' from Figure 5.6, the group had no way of knowing who propagated the changes, who was to receive notification, how the changes would be represented and propagated and what had to be done to realise the propagation. Not to mention what a reasonable time frame for such an action would be. Answers to these questions required clarity beyond the level at which the participants and in particular the designers amongst them, were comfortable. A consequence of this saw more operational goals offered in a way reminiscent of the responses given during the scenario elicitation of Chapter 4.

This raises the important issue of understanding the repercussions of goal decisions as design decisions. It seems all too logical to derive goals to achieve other goals, yet there is not a strong level of understanding as to the impact these decisions may have on the overall system design. For instance in deciding to store user preferences and service subscriptions as a goal, there are strong needs for availability to always respond to the consumer, for performance because the amount of consumers will be very large, as well as for security because it is handling their personal details. Elements of security are present in the decomposition, yet the notions of availability and performance due to large user numbers are lacking. Similarly the ramifications of fulfilling these needs for the overall system remain largely unexplored. For example realising availability may require data refresh and repository mirroring, all of which may require system resources and affect other functions in unforseen ways. It is difficult to maintain the focus required to exhaustively consider all of the issues when decomposing one goal into another. There is an inherent need for truncating the graph at some stage in order to maintain its usefulness. This seems in conflict with the GBR intentions of utilising goal decomposition as the primary means of developing system requirements. As indicated in section 5.3, requirements should be complete and unambiguous (IEEE, 1998b).

During the derivation of the goals in Figure 5.6 several of the participants offered the view that quality attributes were inherent in some of the goals. The suggestion was made that the goals refining 'user self management' implied some well known quality attributes. For instance persisting profiles indicated the need for robustness and protecting profiles related to security (Figure 5.7). When goals occur in a certain context they seem to become clearly related to the quality attributes that had been elicited in the first ATAM exercise (Chapter 4). The interesting point here is that the qualities are not availed until well down the goal graph. Figure 5.7 shows the section of the goal graph, from which the goals implying security and robustness were refined.

The preceding goals such as 'enable user self management', 'increase service personalisation' and 'appeal in a capability (QoE) sense to consumers' can only seen to have tenuous links to these quality attributes. The semantic of a goal does not readily reveal all of its facets, instead its meaning appears to be constructed as part of the refinement process. There is nothing explicit about the need for security to protect profiles in the goal to 'increase service personalisation' or 'appeal in a QoE sense to consumer'. Characterising high level goals as quality requirements, as is the case in the ATAM utility tree, potentially suppresses the refinement reasoning required to establish a meaningful relationship between the two. Similarly the likely occurrence of quality concerns at different levels of abstraction and in different contexts makes them potentially problematic elements when used as a starting point is elucidating system requirements, as is the case with the utility tree in ATAM.

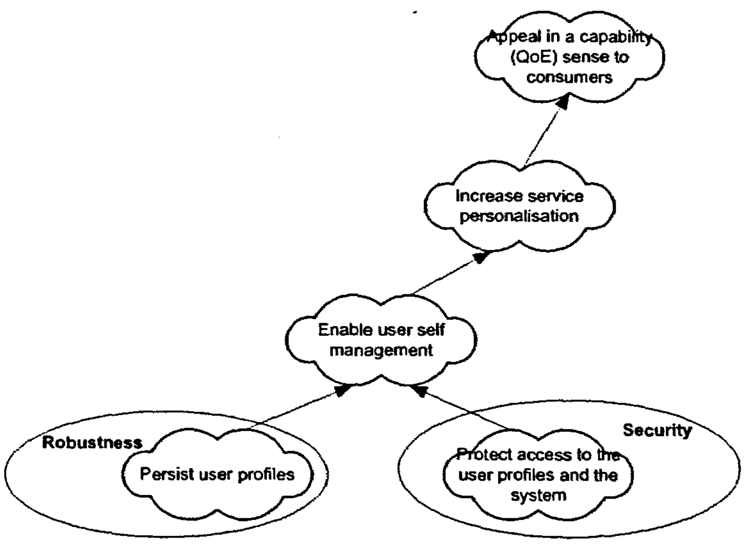

Figure 5.7 - Occurrence of quality attributes in the overall goal hierarchy

The focus until this point on reaching formalisable goals meant that the existing goal graphs all exhibited several layers of refinement but within these graphs very few interdependencies had been identified and represented. Reasoning about interdependencies is critical because it reveals instances where the achievement of one goal is likely to positively or negatively affect the achievement of another. Assessment of these interactions is subjective as the satisfaction of soft goals cannot be established 'in a clear cut sense' (Lamsweerde, 2001; Mylopoulos et al., 1992). The problem with interdependencies is that relationships can be so numerous and differ subtly in semantics that it could become very difficult to establish the significance of having interconnected goal nodes in the manner recommended by the KAOS literature. Instead the interdependent relationships were derived separately from the goal refinement hierarchy by taking each goal at face value and seeing if it held consequence for any other goal. The simple interconnection semantic of 'supports' for positive relationships and 'doesn't support' for negative relationships was specified and the goal graphs examined for interconnectedness.

The relationship was represented firstly in sentences and secondly through goal-relation diagrams such as that in Figure 5.11. Figure 5.8, Figure 5.9 and Figure 5.10 depict the sections of the goal graphs, from which the goals 'Minimise Expansion Cost', 'Enable Fast Service Deployment' and 'Reduce the Operational Complexity of the Network' were taken.

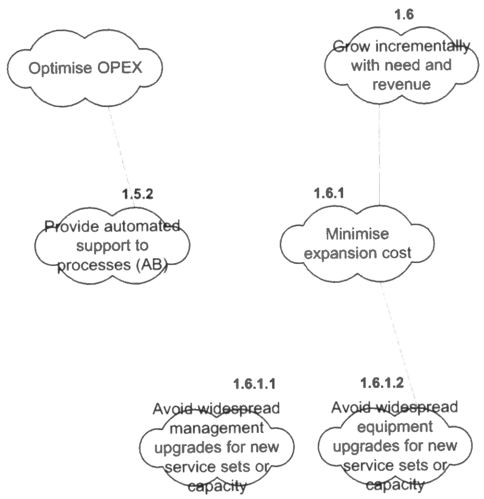

Figure 5.8 - Minimise Expansion Cost – context

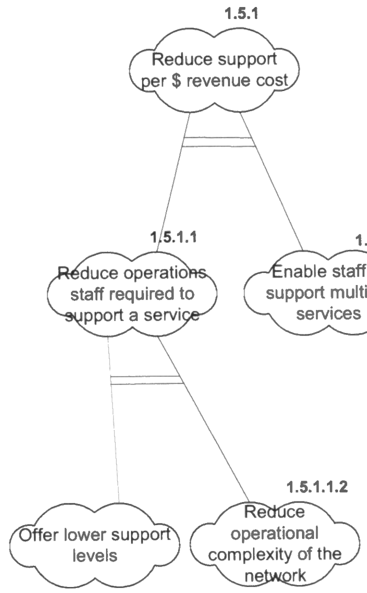

Figure 5.10 - Reduce Operational Complexity of the Network - context

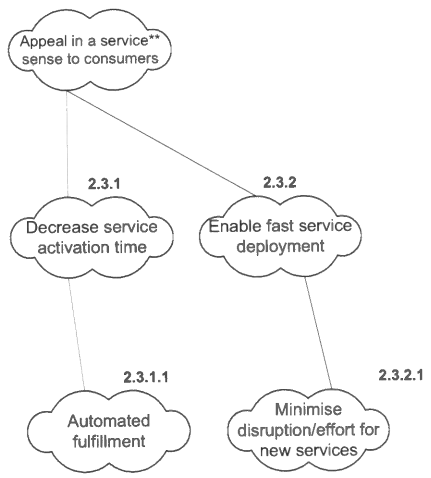

Figure 5.9 - Enable Fast Service Deployment - context

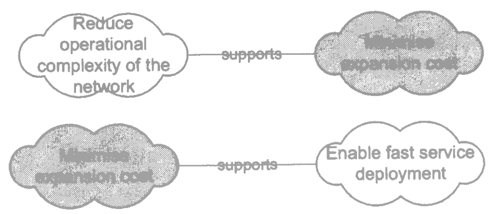

Figure 5.11 - Simple Goal Interconnections

The occurrence of goals in multiple relationships allowed them to be used in grouping goals together. Figure 5.11 shows how the goal 'reduce operational complexity of the network' was seen to support 'minimising expansion cost' and in turn, 'minimise expansion cost' was seen as supporting the 'fast deployment of services'. The existence of the 'minimise expansion cost' goal in both the goal relations enables it to be used as a lynch pin in deriving goal connectedness. In this way, clusters of closely related, supporting goals can be realised. Akin to Lee's stable kernels, these goal sets help simplify considerations early on in system design where goal conflicts can overload designers (Lee & Hsu, 2002). The resulting goal strategy graph incorporating the goal 'minimise expansion cost' is shown in Figure 5.12.

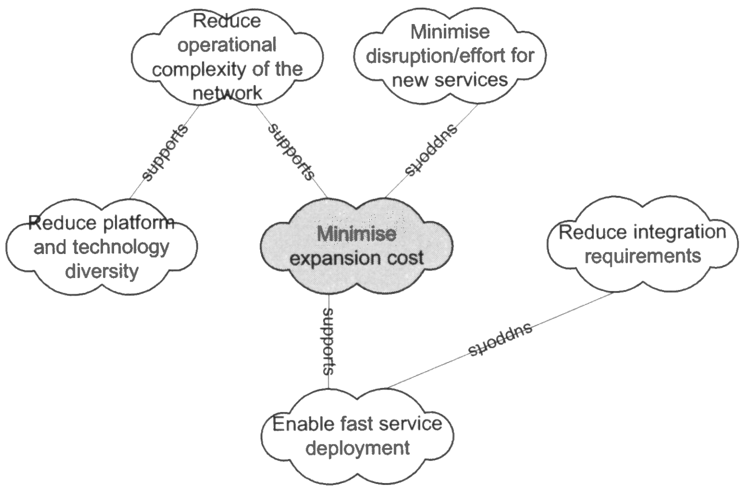

Figure 5.12 - Goal Strategy Graph for 'Grow as you Grow'

Grouping goals like this saw clusters of goals form out of the interactions, permitting the group to readily derive 5 key strategies from the goal graphs. These strategies were 'grow as you grow', 'dynamic management, of resources, performance and security', 'automation', 'convergence' and 'personalisation'. While acting in this way disregards the previous goal refinement structure, the acts of refining and interconnecting goals seem to serve two different contexts. The goal-oriented requirements language (GRL) establishes some precedent for such thinking in offering a multi-step, goal-based process for developing system requirements and solution concepts (Liu & Yu, 2004). The first steps seek to understand the business objectives by modelling them as soft goals and refining them. Once a clear picture of the soft goals has been developed, the later steps of GRL juxtapose the goals with a solution and establish positive and negative relationships, according to whether a solution helps or hinders a goal (Liu & Yu, 2004).

In this way vertical refinement of goals can be seen to be breaking down the complexity of the higher level goals in order to understand their constituent elements, these are purely optative aspects in that they specify capability that currently doesn't exist but is desired. Interconnections (between goals) are related to the context of design where beliefs about the formulation of a design are overlaid onto the soft goals. Therefore the refinement can be seen to search the desired solution space in a bid to build understanding and explore options, whereas the interconnections seek to understand the limitations of a design and acts to constrain the set of achievable soft goals. Exploring the goals first in refining prevents the solution space from being prematurely constrained and breaking up the structure to group goals into strategies avails the designers of a coherence that permits them to look at broad solution concepts rather than each individual goal interaction.

5.7.Revisiting the framework

One of the most prominent framework outcomes from the first ATAM exercise was the relationship between system quality and business strategic concerns. The earlier sections of this chapter established the relationship between these perspectives as indicative of a broader interdependency in design, which is the need to co-evolve the notional problem and solution spaces (Dorst & Dijkhuis, 1995). The incomplete understanding of the business strategic perspective prompted the design team to develop understanding in this area, with the goal-based requirements seen as an appropriate means of structuring and further utilising the goals.

Ultimately the goal-based requirements exercise did not achieve what the group had hoped it would. While important understanding was generated, the original intentions of developing a detailed picture of the strategic perspective and refining it down to system level requirements proved unrealisable. One of the significant barriers to achieving formalisable goals was the complexity of the business strategic perspective. As has been commonly discussed throughout this thesis, complexity comprises both a systems and people dimension (Flood, 1988). Both of these complexity dimensions were as evident within the goal-based requirements (GBR) exercise, as they were in the first ATAM.

The systems complexity associated with number of parts and relationships was highly evident in the breadth and depth of the goal graphs. The extent of these graphs highlighted the potentially unwieldy nature of goals when used to satisfy the need to derive solid technical requirements. In a bid to realise formalisable goals several extensive goal graphs were constructed. However these graphs fell short of producing the necessary entities and concrete constraints that characterised formalisable goals in the KAOS literature. A significant factor in the exercise was a lack of systems clarity, something which appeared to equally affect the scenario exercise of Chapter 4. Even though refinement is meant to incrementally reveal detail, it appears the GBR exercise did not yield a great deal more system clarity than existed in the ATAM workshops. It appears largely infeasible to actively refine from objectives all the way down to solid technical requirements in the one exercise, when only vague solution concepts exist. Problematically, for a goal-based methodology, goals are most useful early in the design lifecycle. Without this bridge between the goals of the business perspective and the requirements of the system perspective, the KAOS methodology loses a lot of its efficacy.

The sheer size and depth of the final goal graphs showed that the soft goals, offered as driving requirements, were lexically simple but conceptually rich. Potentially this is to be expected when looking at goals with a requirements viewpoint, however it is probably better understood by looking at the nature of the GBR task. Whilst the goal graph was exploring what might be termed the conceptual problem space, it was also simultaneously attempting to solve the problem. Joseph suggests there are no problems detached from solutions, as *"we do not experience a need for something we cannot conceive"* (Joseph, 1996, pp 238). Similarly Davis points out defining requirements as 'what' the system must do not 'how' it must do it, falls down during refinement. Rather than maintaining requirements and design, or problem and solution dichotomies, he suggests there is a *"continuum of issues ranging from the most problem-like ... to the most solution-like"* (Davis, 2005, pp 5). Recalling that the design approach is influenced by simply naming what aspects of the situation will be attended to (Schön, 1991), attempts at resolving the design problem are being made from the placement of the first goal onwards.

With every goal refinement, an aspect of the solution is revealed in some way. Likewise the refinement creates a new problem to be resolved in that it represents a new capability that doesn't exist, a way of realising which needs to be discovered. The branching of refinement to follow specific aspects of the problem develops hierarchies of sub-problems and sub-solutions. Joseph suggests when there are interdependencies between the sub-solutions, the whole cannot be satisfied by developing the parts separately. The potential number of solution permutations required to satisfy all combinations of confounds associated with the interdependencies is impossible to address (Joseph, 1996). A good heuristic that is applied to resolve such design situations is often to draw together parts that have a high level of dependency (Joseph, 1996). The concept is similar to that of coupling in software development, where it is used to create high internal cohesion and expose the more significant interfaces. In the case of goals, high internal cohesion means the goals express similar or associated needs, whose achievement supports one another. The coarse interfaces resulting from drawing together goals represent the major antagonistic issues amongst the goals where trade-offs are likely.

Parallels with the construction of the ATAM utility tree are evident in the conduct of the strategies exercise. The ATAM takes a set of high-level quality attributes and refines them into behavioural quality requirements, which can be described as scenarios. Architectural design approaches that are key to the satisfaction of more than one quality requirement are then noted as trade-offs. In this instance the high level requirements were business strategic goals, the refinement took place through goal graphs. In this instance the high level requirements were business strategic goals, the refinement took place through goal graphs. The identification of goals that seemed to support one another were grouped into strategies and cooperative and conflicting set relationships identified. Formulating the 'goal strategies', drew together goals whose achievement was mutually constructive. Mutually constructive indicates that if a system were developed to achieve one goal, it would most likely contribute to the achievement of another goal. The strategies and their contained goals formed part of a revised discipline, imposed upon the design situation (Schön, 1991). This simplified what would otherwise be an enormously complex set of negative and positive interdependencies between goals at arbitrary levels in the goal hierarchy. According to Joseph this achieves an important design step of visualising *"a significant decomposition into approximately independent subsystems"* (Joseph, 1996, pp 235).

Once complete, the goal graphs depict the design moves that took the group from the starting goals to the terminal leaf nodes, mapping out the tree of moves that remark the search through the solution space. Each branching represents a local experiment where the forward repercussions of the move have to be considered as well as the coherence with the existing problem frame (Schön, 1991). However this search cannot be exhaustive as the sheer number of variants in the process make such a task impossible (Joseph, 1996). Whether the constraint of the search is truncated arbitrarily in search of a workable design (Joseph, 1996), or expertly with the confidence of discipline 'virtuosity' (Schön, 1991, pp 104), the end result is an incomplete record of the design considerations. Several of the reflections highlighted the incomplete nature of the goal graphs. The pattern of refinement showed a bias towards functional aspects of the system over metabolic aspects. The lack of stakeholder perspectives other than that of the business also became evident. In addition to this, there were goals that couldn't be placed in the existing graphs, as well as the many goals that were removed or never made it onto the goal graphs.

The decision as to what elements are truncated from the search is based on decisions about 'what is the case' and 'is this good or bad'. The former represents reality judgements and the latter represents value judgements. The process of evaluation is termed 'appreciation' (Checkland, 2000; Jackson, 1988). These judgements can only be made from each participant's personal understanding and intentions. This understanding extends beyond their view of the design as an object of study, but also involves their view of themselves in the situation. It is offered as no coincidence that the chief designer was most adamant about the potential for implicit designing. This aspect of the design was of particular concern to them at the time and any ill-informed decisions in this area would prematurely determine a design issue that needed further exploration. The sharing of these appreciative judgements and understanding amongst the group highlights the need for the negotiation of meaning in concert with the negotiation of competing goals (Bucciarelli, 1988; Sargent, 1994).

Artefacts themselves do not carry sufficient information in order to be understood. The diversity of individual perspectives within a situation ensures meaning needs to be negotiated. The initial withdrawal and creation of an individual goal graph by one participant highlighted the importance of people within a group situation being able to accept each other's reality and value judgements. Following the creation and revision of several individual goal graphs, the group were being asked to approve of artefacts they had no say in the creation of. The withdrawal from the group process and creation of individual goal graphs, as well as the subsequent difficulties engaging the broader stakeholder group indicated that simplicity in language does not pre-suppose the simplicity of the concept. In developing goals rather than system quality concerns, some lexical difficulties of utilising a parochial language set are avoided, however the situation that there are as many individual viewpoints as there are participants and only one end artefact remains. The need for negotiation in design to take place in a social framework is foremost in these events (Sargent, 1994).

The personal nature of these judgements in a group situation elevates the importance of the people dimension of complexity in the GBR exercise. Chapter 4 raised the notion of group perspectives such as those of business strategic and systems quality, as well as the personal perspectives of each participant, termed object worlds (Bucciarelli, 1988). These perspectives influenced the way in which participants contributed, manifesting itself in their language, concepts and interpretations.

Chapter 5 explicitly addressed group perspectives in that it sought to balance existing knowledge in the systems domain through exploring the business strategic one. While the goals showed a strong semblance to the strategic issues from the first ATAM, quality attributes only appeared sporadically in the goal exercises. Systems quality concerns only really began to emerge during the lower level refinement of the goal graphs. There were definite contexts within which goals could be characterised as quality concerns, however they weren't dominant in the graphs. The number of refinement levels between the higher level goals and quality concerns in Figure 5.7 also provided insight into the extent of internalised reasoning required to characterise goals as quality requirements in the ATAM utility tree.

In targeting learning from a business strategic perspective, the GBR exercise shifted focus away from other important stakeholder groups. The soft view of systems adopted in this thesis refutes the single course directed by a 'helmsman' (Checkland & Holwell, 1998, pp 47). Instead there are multiple courses arising from within the system, several of which were briefly explored. The exploration of the customer and network administrator perspectives demonstrated the differences that might arise from entertaining goals of different stakeholders. The lack of these perspectives in the overall exercise is closely related to the view of achieving goals versus maintaining relationships. The cybernetic view tends to believe that goal-setting is a higher order function and that once specified, the function of the system will fall into line with it (Jackson, 1988). The 'soft' view of systems promotes the view that goal seeking is an impoverished view of organisations (Checkland, 2000). The goal graphs may appear to accurately capture what the business owner wants to achieve, however it clearly overlooks the relationships that it must maintain in order to be successful. Of concern is the fact these perspectives were not missed until late into the exercise. Showing how the desire to search for solutions becomes a surrogate for understanding, masking out elements of the situation, in this case diverse perspectives in favour of discovering a design approach.

In assessing the success of the goal graphs it is important they aren't viewed as a means unto themselves. Provided they have significant breadth and depth one might assume they have adequately represented the issues at stake and available approaches. Yet in analysing the outcomes of each of the local experiments constituting the tree of moves, the designer must be cognoscente of the factors that have been eliminated (Schön, 1991). Without recollection of these factors, it may prove difficult in the future to recapture the processes leading to these 'ends'.

Figure 5.13 depicts an evolved framework of ideas, incorporating the reflections of Chapter 5. Notably this framework builds on the final framework of Chapter 4. The use of the framework in section 5.4 above (Figure 5.3), was only used to establish the GBR work in the context of the existing framework.

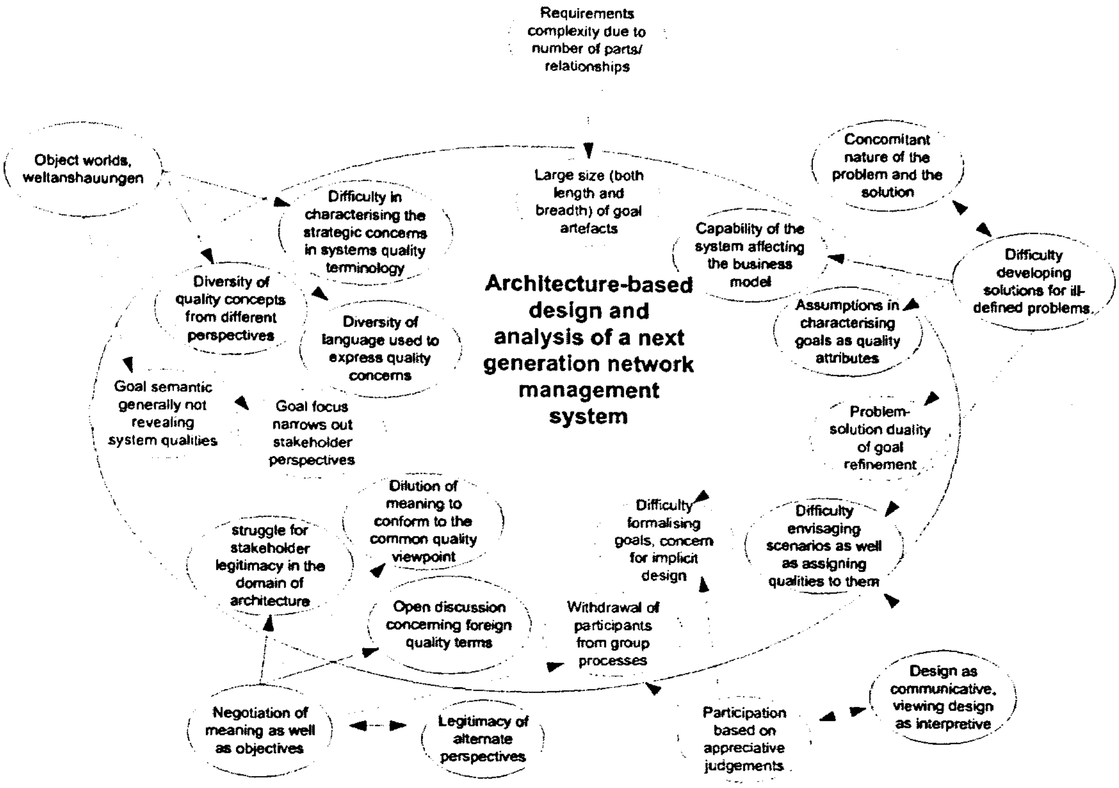

Figure 5.13 - Relationship between the framework and the learning from the second AR cycle

Summary

Chapter 5 recounted the attempts of the project group to come to grips with the apparent complexity of the strategic viewpoint and how it could be reconciled against the need to establish firm system quality goals. Goal-based requirements were presented as a promising way of both addressing the need to model the strategic viewpoint and transfer that knowledge into system design activities. It also showed potential to alleviate some of the problems experienced in the first ATAM exercise, such as the disparate nature of the strategic and systems quality viewpoints and the difficulty communicating between the diverse stakeholder perspectives. The application of the KAOS methodology for GBR requirements tested these notions and the learning was represented in the evolved research framework of ideas.

Overall the exercise did not achieve what it set out to and eventually the group resorted to developing its own artefact that used as an output from the process. While this could be viewed negatively, the learning attained during the process pointed to the fact this was a useful exercise and that it is unlikely that the expectations the group had at the start of the process would be realised early on in such a complex design situation. Although goals present themselves in relatively natural language, they are deceivingly complex because they abstract so many facets within their simple terms. The extensive refinement that took place and its subsequent inability to realise goals that could be formalised, demonstrated the potential pitfall of attempting to refine goals into system requirements.

Goals provide a tangible way of capturing useful strategic knowledge at the start of a project where the problem is ill-defined. As indicated by the qualities of systems age synthesis, understanding is valued over structure at this stage. However GBR frameworks like that of KAOS tend to judge their outcomes by whether formal requirements are achieved or not. It was demonstrated that to achieve such outcomes implicit design choices imposed themselves upon the situation. The comprehensive nature of the GBR methods means they are only likely to meet maturity as the design nears maturity, therefore the aspirations of what can taken from such an exercise need to be tethered to the extent of understanding. In this instance it proved very useful to come away with a picture of the strategic issues and a set of strategies whose application would largely account for the system needs implied by the strategic issues. These strategies and goals contained within them would drive design work from this point and form the basis of future analysis. Chapter 6 provides insight into the use of this knowledge within another ATAM analysis, following a significant period of project design activity.

6

The Pronto ATAM

Introduction

Chapter 4 presented the first significant action research cycle, describing the application of the architecture trade-off analysis method (ATAM). The ATAM was used to develop an understanding of the problem situation and published solution approaches. The analysis outcomes established a broad interdependency between the business strategic and systems quality perspectives. Stemming from an incomplete understanding of the strategic perspective, Chapter 5 examined the use of goal-based requirements (GBR) to model the high level system drivers and develop their association to system concerns. Following this early analysis work the project moved into a sustained period of design activity focused on creating an initial system architecture.

Chapter 6 establishes the need to challenge the conceptual decompositions resulting from early design work. ATAM is the preferred method of 'challenging' the architecture, though the difficulties encountered in the previous chapters suggest it is vulnerable to the uncertainty which abounds in the design when 'considered as a social process' (Bucciarelli, 2002, pp 221). The mitigation that the project hadn't developed its own clear design is offered with the expectation that greater understanding will yield better analysis results. The remainder of Chapter 6 presents the ATAM activities and reflects upon the outcomes with respect to the theoretical framework of knowledge established in Chapter 4 and evolved in chapter 5. The structure of Chapter 6 is shown in Figure 6.1.

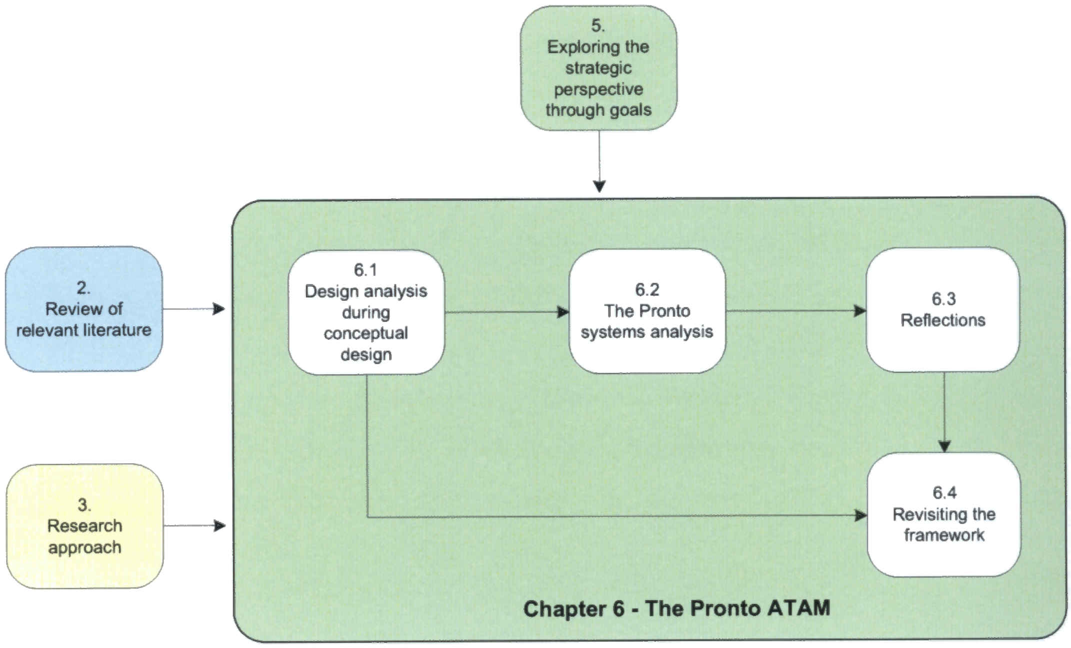

Figure 6.1 - Structure of Chapter 6

6.1.Design analysis during conceptual design

Having developed a considerable understanding of the problem and solution spaces, the project team refocused on producing a preliminary architecture, or high-level system design. As is commonly the case in early systems design, the high-level architecture focused on showing the assignment of aggregate function to components as well as the configuration and connectivity of these functional components. The decomposition at this stage is purely a conceptual one (Pahl & Beitz, 1984) and needs to be continually tested as design progresses.

> *"During the generation of the conceptual components, decisions made during the definition of the conceptual subsystems may be revisited in the light of new information. This type of re-investigation in light of better understanding is a characteristic of all designing efforts including those using the ABD (Architecture Based Design) method."* (Bachmann et al., 2000)

In the context of architecture-based design, revision is enabled through the iterative application of architecture-based analysis throughout the design lifecycle (Bachmann et al., 2000; Bosch, 2003). Chapter 4 introduced a dichotomy of architecture-based analysis methods, split amongst the groups of 'questioning' (qualitative) and 'measuring' (quantitative) (Abowd et al., 1997). The need to develop specific models of system behaviour to facilitate the quantitative aims of measuring techniques discourages their application early in the systems life cycle, where quantifiable information is hard to come by and often incomplete (Gurp & Bosch, 2000). During this time questioning techniques like ATAM^{SM} are advantageous because they are coarser grained and aren't as susceptible to the pitfalls created by incomplete system models. Such methods are aware of the fragile reality that exists and have adapted themselves to cope by allowing for information gathering and consultation periods. Bass even suggests that the early application of architecture-based evaluation makes an 'excellent discovery vehicle' (Bass et al., 1998), helping to avoid the development of conflicting system properties that often arises when requirements creation is isolated from early design activity (Bass et al., 1998).

However experience in this research project has shown that the problem is not just about information to be gathered and communicated, but is also about the uncertain nature of the information itself. Chapter 4 demonstrated the potential for different stakeholder perspectives to impinge upon a common understanding of the system quality requirements and their relationship to the driving strategic need. Uncertainty about the system also impeded the group's ability to specify usage scenarios beyond the well established operational specifications of a management system. Throughout the first ATAM the negotiation of meaning and objectives from diverse stakeholder perspectives proved to be a key issue.

Chapter 5 re-enforced the need for negotiation amongst perspectives with significant issues arising from the inability to reach consensus on group artefacts and the unitary viewpoint of the exercise. Insight into the scenario problems was also achieved, showing assumptions would need to be made about the form of the solution in order to specify formalisable goals. Formalisable goals are analogous to scenarios in that they specify the intended behaviour of the system given a starting state and specific environmental trigger (Lamsweerde, 2000). The lack of system detail preventing the specification of formalisable goals is symptomatic of the problems experienced specifying scenarios in Chapter 4. In the same way designers are unwilling to embellish the system solution enough to realise formalisable goals, users will find great difficulty in specifying interactions with a system that is insufficiently detailed. Brooks notes *"it is really impossible for a client, even working with a software engineer, to specify completely, precisely, and correctly the exact requirements of a modern software product before trying some versions of the product"* (Brooks, 1987, pp 11-12). Although such 'trying' has very physical connotations and Brooks went on to endorse rapid prototyping, systems architecture still engenders both behavioural and usage aspects of a system, the clarity of which will effect how stakeholders envisage their interaction with it.

The problems envisaging usage aspects of the system hold particular consequence for architecture-based analysis methods employing scenario-based techniques. Scenarios form the cornerstone of the questioning methods (Dobrica & Niemela, 2002), which as mentioned previously are particularly suited to the earlier stages of design. Instead it appears such methods may be better suited to middle stages of design, once a degree of architectural elaboration has taken place and an architecture exists (Abowd et al., 1997), all be it in varying stages of completeness. Although candidate architectures were used in Chapter 4, the first ATAM could be viewed as preceding the elaboration of the project team's own architecture, which Abowd advocates is necessary for scenario-based methods.

Architectural detail and level of maturity vary considerably between the tentative first cut and the detailed final design. Therefore architecture analysis can be seen take place across a broad spectrum of architecture maturity (Abowd et al., 1997), something the ATAM is suggested as being suited to (Dobrica & Niemela, 2002). Case reporting literature from the SEI focuses on more mature architectures that have a reasonable level of system detail such as the technology makeup of the system and the functional responsibilities (Clements et al., 2002; Kazman et al., 2000). The successful outcomes of these case reports support the application of the ATAM to mature architectures but leaves the question of its efficacy on less mature architectures unanswered.

The following chapter discusses the use of the ATAM on the first release architectural design for the collaborative project's management framework. The immaturity of the architecture relative to that required of a final design should shed some light on the effectiveness of using a scenario-based method of analysis like ATAM on early system designs. The lapsing of over a year in the project and elaboration of the architecture since the first ATAM would also provide a contrast with the use of the method at the beginning of the design lifecycle as recounted back in Chapter 4.

The codename 'Pronto' is introduced in this chapter and will be used to refer to the project's prototype next-generation network (NGN) management system, from here on in.

6.2. The Pronto Systems Analysis

The Architecture-based Trade-off Analysis Method (ATAM) was presented in Chapter 4. The lack of access to the original designers and requirements focus of the original exercise prevented that analysis from progressing beyond step 5. However the Pronto ATAM made use of the full set of ATAM stages, which are depicted in Figure 6.2 (Kazman et al., 2000).

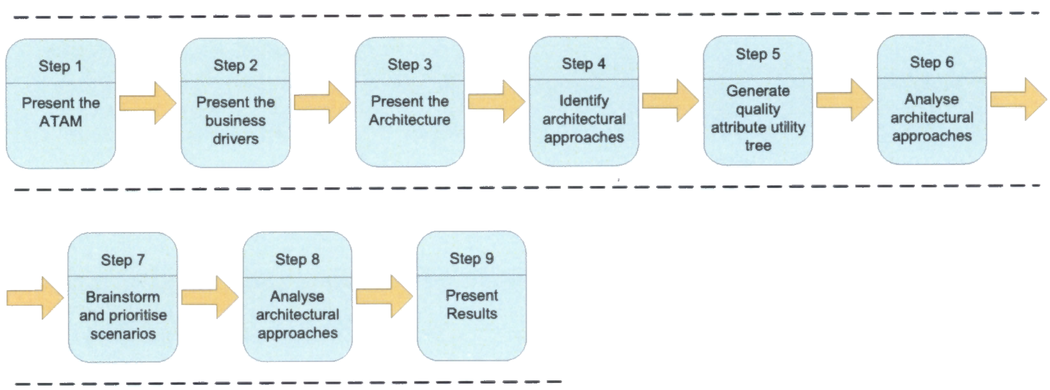

Figure 6.2 - ATAM Phase 2 steps (Kazman 2000)

The process commonly utilises two phases as discussed earlier in Chapter 4, separated by a break of a few weeks. The break is designed to assess the outcomes of the first phase and further develop the system architecture *"models that will give the evaluators and the architect sufficient insight into the architecture to make the Phase 2 meeting more productive"* (Kazman et al., 2000, pp 40). The assumption being that the models developed by the architects are generally not capable of providing the sorts of insight required by the ATAM. While this may tend to be the case in a commercial consultancy situation where the analysis team is called in to bring the architecture and business teams together, no such clear separation of roles existed for the Pronto analysis. Though the design leader for the architecture and the analysis facilitator were indeed different people within the team, both were commonly involved in each other's work. Adding to the participant's involvement throughout the project, the initial ATAM exercise presented in Chapter 4, can also be considered as performing the role of a classic ATAM phase 1, with the break between that original exercise and the current analysis acting like an extended consultation. Consequently the second phase of the ATAM will be used as the model for the analysis of the Pronto architecture.

The traditional planning exercises of gathering information from disparate sources, etc that usually take place were made easier by the personnel situation. As a result the planning for the Pronto ATAM revolved around administrative duties and ensuring the workshops ran smoothly. Important factors included ensuring the presence of key team members and availability of important artefacts. Although the ATAM suggests a two day time frame for the second phase, they also note that it needn't be slavishly followed. The resources were available, and in a somewhat flexible manner suggesting the analysis progress would be dictated by the group. In total four workshops were conducted to complete the Pronto ATAM.

The first workshop focused on drawing together all of the business strategic artefacts that had been accrued during the project. They were re-structured to account for overlap between multiple exercises and shaped according to the ATAM information template. Information associated with this template included the 'business environment, market differentiators, driving requirements, stakeholders and how the proposed system meets those needs/requirements, business constraints, technical constraints' as well as 'desired quality attributes and what business needs these are derived from'. This task was made all the more easy by having conducted the first ATAM and GBR exercises, many of the artefacts were able to be directly reused.

The second workshop saw the architecture presented to the group. The main system views presented to the group were the context view (system of systems decomposition including external entities) and the functional view (both first and second levels of refinement from the systems view). The early stage of the design prevented the presentation of more detailed views such as those specifying assignment of function to physical devices and resulting execution paths. Following the presentation of the architecture, the architecture was examined for the use of common styles. The predominant styles found were those of a blackboard style, a layering style, a tiered style and a client-server style. The styles and their implications for the design were discussed only briefly before the group moved onto attempting to build the attribute utility tree. The rest of the workshop was spent attempting to construct the utility tree, which as discussed in the reflections didn't prove as straight forward as first thought.

As a consequence of the difficulty encountered constructing the utility tree, a significant portion of the third workshop was also devoted to further building the tree. In particular the operational aspects of the system like scenarios designed to test the quality goals were proving hard to elucidate. Different strategies were employed to grant context to the scenarios and provide clarity for the workshop participants.

The final workshop re-presented the attribute utility tree, as not all members of the group had been present during the third workshop. The lead designer for the Pronto system was then asked to demonstrate how the scenarios would be enacted by the system. The description was used to develop the points of sensitivity, which are essentially architectural decisions designed to satisfy a particular quality constraint. Trade-off points are denoted as decisions responsible for multiple quality constraints. Architectural risks are also identified focusing on important design architectural decisions that are as yet undecided.

6.3.Reflections

6.3.1. Presenting the business case

The first workshop focused on presenting the ATAM itself and what is commonly termed the 'business case' for the system. The business case addresses issues such as the organisational and market environments that have led to the business need, as well as the proposed solution that will meet it. Importantly for the ATAM, the desired quality attributes of the system are also presented and justified against a driving system need. This simple causal attribution carries strong consequences for the ATAM outcomes as these qualities form the backbone of the utility tree, which largely dictates the issues that will be focused on in the analysis.

The bridging of the business strategic and systems quality perspectives previously proved to be a non trivial exercise, with Chapter 4 elaborating the associated difficulties. Consequently, Chapter 5 used the KAOS goal-based requirements (GBR) methodology to extensively explore the business strategic perspective and derive a relationship between it and the systems perspective. The goal refinement and subsequent association to form strategies yielded a neatly abstracted set of system needs. However the refinement of the goal graphs didn't readily provide a set of quality concerns that would satisfy the higher level goals. Extensive refinement of the goal graphs revealed systems qualities in some contexts but they were by no means a dominant feature of the goal graphs. This precluded clear associations being made between the goals and the quality attributes that would be most critical for their achievement.

Unable to derive clear system quality attributes from the outcomes of the goal-based requirements exercise, the group turned to the quality attributes elicited in the first ATAM exercise and sought to meaningfully associate them with each goal (termed a driving system need in ATAM). Each potential association was discussed in an open floor fashion so that general concurrence was required for an attribute to be considered a legitimate candidate for satisfying a system need. In doing so the quality attributes were derived taking lead from the meaning of the driving requirements. For example 'dynamically manage' was perceived as being supported by the system quality attributes of 'timely performance', 'growth' and 'security'. Figure 6.3 below shows the quality attributes and the driving system needs they were derived from.

Driving system need	Quality attributes
Automate	Timely Performance
Grow-as-you-grow	Growth
Dynamically Manage	Timely Performance, Growth, Security
Converge	Growth
Personalisation	Growth
Federated Management	Growth, Openness, Maintainability, Simplicity
Flow Through	Timely Performance, Openness, Simplicity
Service Management	Timely Performance, Growth, Simplicity

Figure 6.3 - Main system quality attributes and the driving system need they are derived from

A noticeable aspect of the business case presentation was the disparity between the amount of information put forward, compared to the amount of information utilised within the analysis. Aspects from the business case such as the business context and history were discussed openly during design meetings and undoubtedly helped shape the contributions to the goal graphs from Chapter 5. However they weren't formally integrated into the derivation of the utility tree. In the same way that quality attributes effectively condense behavioural meaning into single words, the driving system needs tightly condense a significant amount of business strategic knowledge.

6.3.2. Presenting the architecture

The focus on the first workshop was on completing step 2 of the ATAM, which is to formalise the business position. The second workshop undertook the next three stages in the process which are to present the systems architecture, identify architectural approaches within the design and finally to create the attribute utility tree.

Given the fairly immature nature of the systems design, the architecture presentation focused on two classic views, the system context view and functional decomposition. The main thing of note from the presentation itself was the difficulty in deriving detailed operational descriptions of the system at such a high level of abstraction. A lot of the scenario traces incited questions from the group but were unable to be resolved in detail. The principal designer presenting the architecture answered a lot questions in a way that offered design permutations, such as 'if we distributed this functionality here, or moved this component here, then the flow of execution would look like…'. The designer noted that many of the conditional answers were based upon the importance of understanding the effectiveness of the policy aspects of the system before proceeding with more detailed design activity. All of the permutations of the architecture were based upon the designer's understanding of the trade-offs inherent in the design.

The situation is the inverse of the Chapter 4 ATAM which was held early in the project lifecycle and the focus was on the clarifying the business needs and trade-offs in order to formulate the problem. The business stakeholder is the focus during this initial stage, as the designers and architects seek to understand the business concepts. During the Pronto ATAM, where the design has progressed appreciably, the designers became the focus of inquisition as they understand the intricacies of the design and its consequences. In each case communication from one perspective to another is critical for stakeholder achievement. Resolving issues in either perspective demands that the other be equally well considered and designed, with resolution to design issues in one perspective concomitant on understanding in the other.

As was the case for the operational aspects of the system, the abstract nature of the design also made the task of identifying architectural approaches more difficult. However on discussing the architecture with the designer, some styles became apparent. The design made use of a blackboard style to manage policy distribution and persistence, a layering style to help maintain functional coherence and longer term growth, as well as a client–server style utilised to mediate communication between the service management component and the distributed policy managers. The designer was also able to present on the aspects of the system that they saw as key for ensuring the quality attributes shown in Figure 6.3. Though as with the scenario traces, they commonly offered different answers according to how more detailed design decisions might affect the qualities.

6.3.3. Constructing the utility tree

Moving on from examining the architectural approaches the group turned its attention towards building the attribute utility tree. The ATAM proposes the utility tree be decomposed into 4 layers. Occupying the apex of the tree is the quality of utility conglomerating all of the constituent quality concerns in the notion of overall 'goodness' (Kazman et al., 2000). Under the utility node are more specific quality factors, generally spoken of as quality attributes, non-functional requirements (NFR)s (Bosch & Molin, 1999) or measures of effectiveness (MoE) (IEEE, 1999). In the context of the MEEN project, these were the quality attributes shown on the right-hand side of Figure 6.3, elicited during the quality attribute workshop of Chapter 4. These quality attributes are then refined into sub-factors that are based on characterisations of the quality attributes. An example of which is the performance attribute which can be reduced into sub-factors of throughput, delay and loss(Kazman et al., 2000). Having derived the top 3 layers of the tree, scenarios are used to express the qualities in a concrete way so that the stakeholders can prioritise them. Figure 6.4 shows the example of a user picking up a phone handset and receiving a timely dial tone as a concrete example of delay constraints for the Pronto system.

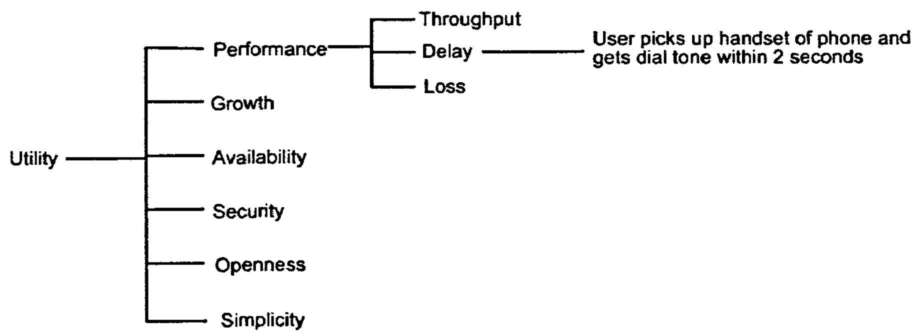

Figure 6.4 - Attribute utility tree structure

The utility tree, down to the sub-factor level was displayed on the meeting room whiteboard and the group began to brainstorm scenarios for the leaves of the utility tree. While a few scenarios sprang to mind quite quickly, after only a short space of time the group at large began to find the task difficult. The difficulty appeared to arise from the lack of context inherent in the 3 upper layers of the attribute utility tree. Although the quality attributes had been derived by the group, there was little mental stimulus to remind them how these qualities had been derived. Most quality attributes are too amorphous to be understood outside of the circumstances in which they were envisaged.

"...quality attributes do not exist in isolation, but rather only have meaning within a context. A system is modifiable (or not) with respect to certain classes of changes" (Abowd et al., 1997, pp 9)

The generic nature of the characterisations may decompose the quality attribute but the resulting quality sub-factors did not appear meaningful to the group. In a system as large and complex as a network management system, there will be numerous time (latency) constraints required for all different types of operational reasons. The analysis must identify the constraints that are of consequence for the driving system need (goal), which gave rise to the quality requirement. Although the driving system needs and quality attributes had been associated at the start of the exercise, the association linked two artefacts that had been developed at separate stages in the project. The participants were aware of the prior contexts in which these artefacts were created and tried to recapture it. In a bid to grant more contextual relevance to the derivation of scenarios, the group expressed the desire to model the driving system needs within the utility tree. The context provided by including the driving system need would prompt recognition of why the quality attribute was important to the system and aid in identifying appropriate scenarios. An example from the altered utility tree with a performance scenario is shown in Figure 6.5.

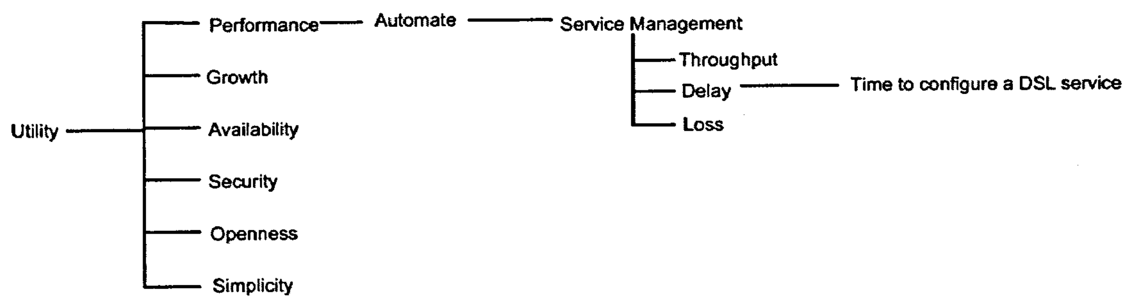

Figure 6.5 - Utility embellished with a perspective on the driving business requirements

Performance was traced back to being important to the need for automation as contained in the associations of Figure 6.3. Following the identification of automation as exacting strong performance demands upon the system, there is a need to understand the elements of automation that grant the perception performance should be a strong driver. In the Pronto system an important part of automation is concerned with service management and provision. A close relationship now exists between the aspect of the system that is strongly dependant upon performance and the operational use of the system that will test it. A scenario was then able to be created that tested appropriate performance aspects of the system associated with automation and service management. In this instance the automated fulfilment of a DSL service request from a subscriber.

Structuring the utility with extra context information from the driving system need proved to assist the scenario generation. However the method used to do so, created issues for the group. Some participants identified overlapping concerns and were worried the group was drawing back to the same set of issues under different banners, whilst the chief designer again voiced concern about how the scenarios tended to imply more system detail than existed. The team also acknowledged that the lack of system detail made it difficult to envisage scenarios other than abstract use cases. Having experienced sustained difficulty in developing an attribute utility tree that was satisfactory to the group, the third workshop focused on how to resolve these issues.

In discussing a potential reason for the group's difficulty developing the utility tree, one of the participants raised the point that they were struggling to recall the reasoning behind the association of the quality attributes and driving system need. They gave the example that they weren't sure why 'timely performance' was given as a requisite requirement for the driving system need of 'dynamically manage'. In order to correctly refine the performance quality factor there is a necessity to make sure the dynamically manage need was represented in the scenarios. This requires understanding why 'timely performance' and 'dynamically manage' are related and recalling the rationale behind the relationship. The arbitrary nature of the association made early on in the ATAM exercise, shown in figure 3, didn't give strong consideration to recording the rationale behind the relationship.

The earlier point about the amorphous nature of quality attributes rings particularly true here. There is nothing inherent about the need for quality attributes in a given system. There is not some natural relationship between automation and service management and performance. Instead there is a carefully derived need for automation, which incorporates the desire to automate service fulfilment, of which, the user getting a DSL service is one example. The quality requirements arise from the confluence of the shape of the design and the goals of the client. The driving system need only exists in the background of the utility tree exercise. Once the utility tree exercise moved away from this understanding and refocused on refining quality attributes, the task became directionless. It took some time of potentially misguided effort before the problem was identified. Importantly this showed that scenarios do not create the context for the quality attributes, but rather rely upon existing contextual understanding for their formulation.

Tracing the way in which the association had originally been derived, there appeared to be a logical hiatus between the quality attributes and driving system needs. The driving system needs had been taken from the outputs of the goal-based requirements exercise, whilst the quality requirements came from the quality attributes workshop(s) held during the ATAM of Chapter 4. The question arose as how to blend these two artefacts so the quality attributes are meaningful in terms of the goals and can be refined to properly test them. In order to bridge the hiatus, a matrix was established with the quality attributes down one side and the goals down another, as shown in Figure 6.6.

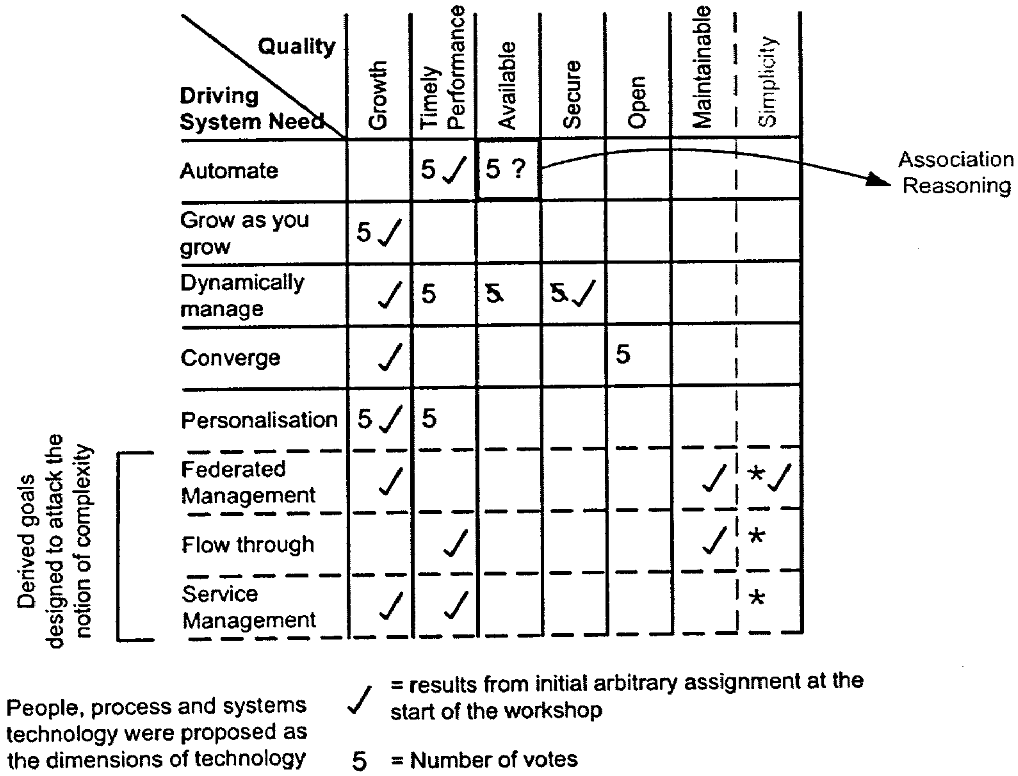

Figure 6.6 - Goal- Quality Matrix

The associations were done one goal at a time, where goals were on the vertical axis (forming rows) and then considered with respect to each quality attribute that formed the horizontal axis (forming columns). If a dependency between the goal and the quality attribute was seen to exist then the number of votes for that dependency were recorded for the stakeholders present and the reason for the association documented on the whiteboard. The composition of the driving system needs, which were effectively the 'strategy' goal groupings from the GBR exercise, played an important part in supporting the association. The GBR artefacts helped provide the context that was suggested above as being critical to the derivation of appropriate scenarios.

From the difference between the placement of ticks and numbers in Figure 6.6 it is evident that group opinion had changed from the initial assignment of qualities to the more explicitly reasoned matrix approach. This helps to demonstrate the uncertain nature of attributions that are made during the analysis process. The outcomes of neither exercise could be considered contentious, with the group generally able to reach consensus on the associations in both instances, yet the end results clearly differed. In this instance the difference was significant enough for one of the participants to note that where the quality attribute of growth (representing qualities like scalability, evolvability, modifiability) had previously been the dominant concern, timely performance now seemed to have surpassed it. Throughout this uncertainty the GBR artefacts provided an important point of reference from which to base ideas. An explicitly reasoned business perspective definitely helps to underpin the reasoning behind the utility tree.

The matrix of Figure 6.6 also bears witness to an issue that had to be confronted when the associations between the quality attributes and driving system needs were being made. The 3rd row down representing the qualities which are important for achieving 'dynamic management' show two qualities that were initially considered by the group but rejected later on. The reversal of this decision was based upon the realisation that the group were including what could be considered derived attributes. As suggested by one participant, there was a distinct danger of creating a series of 'motherhood' statements.

Taking the example of the goal 'dynamically manage', the group decided initially it had natural timely performance needs. It was then suggested that the availability of the system had the potential to impact the timely performance and in turn the availability could be adversely affected by security flaws. The performance concern is one of born of the operational need to possess enough system capability to deal with the task of actively managing carrier network services down to an individual user level of granularity. Whereas availability and security are necessary qualities that the system must be seen to possess in order to support the system in realising its performance objectives. These 'ancillary' (derived) attributes were more the result of interaction between quality attributes, than from a perceived need of the associated goal. Consequently, accepting such qualities as relevant to goals, risked the situation where all qualities were relevant to all goals because of natural interdependencies. However considering these as 'ancillary' concerns on this occasion highlights the singular perspective from which the system was being viewed. Chapter 5 discussed the possibility for different stakeholders to have different goals for the system. These different goals could naturally invoke different quality concerns. The goals developed from the perspective of the network administrator spoke very strongly of the need for availability, as well as monitoring the health of, and trouble shooting any faulty equipment. Attributes labelled as ancillary from one perspective can quite easily represent core concerns from another perspective.

The matrix in Figure 6.6 also shows a set of different coloured driving system needs ('federated management', 'flow through' and 'service management') as well as a differently coloured quality of 'simplicity'. These entries represent additions to the exercise requested by the lead designer, who reasoned simplicity was a necessary quality in order to realise the driving needs of federated management, flow through and service management. The notion of simplicity represents a resolution to a problem encountered whilst developing the design of the policy-based aspects of the Pronto system. In designing the policy system, the lead designer noted that a fine-grained policy solution would overly complicate the management approach. The result would be a more fragmented framework that couldn't easily adapt to new service needs. On this occasion a participant had brought a known trade-off into the process, seeking to air the concern in front of the client and using the ATAM as a forum to do so. Importantly, the issue was one that the design team couldn't resolve alone because the aspects of service expansion and granularity related directly to the strategic goals of grow-as-you-grow and personalisation respectively.

The introduction of these concerns proved problematic for the ATAM facilitator. Later analysis showed that the 3 new driving system needs introduced by the designer showed strong similarity with goals contained within the existing set of goals. At the time, these similarities were difficult to discern, the main reason this overlap did not cause problems is that these introduced parts of the utility tree exercise received very little attention. The group as a whole did not quite grasp the design concern of simplicity and left it up to the lead designer to account for this during the exercise.

Despite the matrix exercise yielding some further understanding on the relationship between the driving system need and the system quality attributes, the group still had problems brainstorming scenarios. Although the previous attempts at creating scenarios had yielded some operational examples of system use, the number was small compared to the complexity of the system and took significant effort to actually produce.

In a bid to create a more free-flowing exercise that yielded a set of scenarios commensurate with the system's size and importance, the group decomposed the scenario exercise. The ATAM literature suggests that the utility tree scenarios are comprised of 3 main elements, the stimulus, the environment and the response. Looking at the structure of the scenarios the most prominent component appeared to be the response measure. While the other components are of equal importance in understanding the scenario, the response measure is the yardstick by which the system is judged. For instance with the usage scenario where the 'user presses start on a streaming video and receives an image within 400ms', the user stimulus is an important facet of the scenario but the focus is on determining the time taken for the system to present the user with a video on their viewing appliance. The response measure and its acceptable bounds are the criteria by which the system is judged. As has been previously raised, quality attributes are logical containers for classes of very real expectations that can only be expressed in observable properties of the system.

Focusing on the response measures allowed the group to detach themselves somewhat from the specifics of the previous method. This simplification permitted them to think more broadly about the variety of measures they saw as appropriate for assessing the success of the system. The shift in focus from carefully deriving scenarios appropriate to the qualities and driving goals, to more arbitrarily nominating measures of system effectiveness helped accelerate the process of deriving the attribute utility tree. The generic form of the measures meant that appropriate scenarios outlining the system environment and stimulus would have to be derived, though this task proved easier knowing the focus of the scenario.

On reviewing the utility tree scenarios and the system measures they were based upon, the group also noted the potential for confusion to occur between the properties of the system being developed and the properties of entities affected by the system. Decomposing the quality attribute 'availability' in the utility tree, the group nominated the measure of 'time to take a rogue device offline'. In terms of networking, rogue devices are those considered not to be functioning in the interests of the carrier. For instance they can be routers with incorrect routing table entries and whose interaction with peers causes routing loops, similarly they can also be servers which have been hacked and are acting as proxy devices for parties other than the carrier. In this instance, the availability of the device and network service is directly affected by the ability of the system to detect and recover from the incorrect functioning, a seemingly valid test of the systems availability. However the availability of the services the network management system enables and monitors need not necessarily be considered to test the availability of the system itself.

Here the situation arises that the system being created has a set of properties, but it also imbues a set of properties into the network. The coupling between the management system and the entity it is designed to manage (the network itself,) is almost too difficult to discern. Network elements act in some capacity as part of the management system, such as when they are setting marking policies and updating routing table entires, yet their normal data forwarding and tagging functions form the cornerstones of network functionality. Further uncertainty about the divisions and bounds of the system arises when considering the different viewpoints involved. The carrier has to be able to operationally differentiate between tasks, however from the consumer's perspective there is no such differentiation. They don't perceive loss of service availability as being differentiable between a carrier's systems. The coupling between the 'systems', as it were and the different perceptions of the system, suggest a tension between maintaining the scope of a system and ensuring all key stakeholder perspectives are considered.

Whilst the exercise was able to be successfully followed through, its success was judged quite differently according to the different stakeholders. The presentation and discussion of the architecture proved quite useful to the participants as it informed them about how the design was progressing and what the architecture looked like. A communication link re-opened, which had somewhat closed following the early ATAM and requirements. However most of the issues that were identified as outcomes, such as key points of sensitivity, trade-off points and risks, were evident to the participants before the ATAM workshops. The lead designer expressed their concern that the architecture at this stage was not detailed enough to really benefit from rigorous analysis. A point echoed by the chief architect who offered the opinion that the maturity of the design did not give them confidence in the outcomes. They suggested the 'implementation architecture' had not been adequately addressed, as it is unwise to logically derive a system and then throw enough computing technology at it to achieve the desired result.

The discussion of styles in Chapter 4 has already raised the importance of implementation details to understanding system behaviour. System architectures can have several implementations, which exhibit different behavioural characteristics. The Cost Benefit Analysis Method (CBAM) method (Kazman et al., 2002), which seeks to resolve the post-ATAM trade-offs and the GBR techniques, which seek to resolve goal-level trade-offs, both seek to make choices between design strategies. However a wise strategy at an abstract level may prove otherwise once implementation details and limitations are considered. The problem facing systems design is that the time taken to develop this detailed understanding is significant. Knowledge at this level will also feedback into the business goals, potentially forcing change in this area. Likewise the environment that gave rise to the business need is constantly changing and providing another source for revising goals (Truex et al., 1999). The industry partner was constantly introducing new issues and aspects to the problem that occurred in their business environment. This flux makes the activities of design and analysis extremely challenging. However to the credit of analysis it at least forces artefacts to be developed that clearly articulate previously implicit considerations. This provides solid grounds upon which to either enact or push back against change.

6.4. Revisiting the framework

The reflections of Chapter 5 revealed a lineage of problems eliciting use case scenarios for the NGN management system. Both the scenario brainstorming in Chapter 4 and the formalisation of goals in chapter 5 struggled to formulate detailed usage aspects of the system. One of the concepts used to understand these early difficulties was that design artefacts perform an extremely important communicative function between the design and user perspectives. The user community envisages the system as-used through their interpretation of these artefacts (Galle, 1999). Consequently the lack of system detail could be seen as inhibiting the communication process and ultimately the ability of the stakeholders to perceive system use cases.

However the difficulty of the stakeholders in developing use case scenarios for the Pronto ATAM despite being presented with the architecture, suggests it is more than the lack of system clarity preventing progress. Kroes offers insight into these problems by suggesting that technical artefacts cannot be understood solely by their physical structure (Kroes, 2002). The function of a system needs to be understood with respect to both its structure and context of human action. In the Pronto design situation, the structure is the domain of the designers and the context of human action is derived from the intentions (goals) of industry partner. This clearly complements the earlier experiences, where co-evolution between stakeholder perspectives in the design situation became an important theme. The result is a dualistic nature, which can be explained in a terms of Simon's 'inner' and 'outer' environments. *"Looked at from the outer environment, the artefact presents itself primarily as something, whatever its inner environment, that fulfils a certain goal, purpose or function…Looked at from the inner environment the artefact is described as some kind of physical system"* (Kroes, 2002, pp 292). The dual aspects are the domain of intentional human action and domain of physical structure.

These domains align naturally with the user and design actors presented by Galle as being on opposite sides of the communication performed by design artefacts (Galle, 1999). Having different world views (Weltanschauungen (W)), these actors view the design artefacts from two very different contexts. The context of design is focused on structure and how to construct a system that realises a certain function. Whereas the context of use is focused on the operational context of the system, where the *"function of the artefact in relation to the realisation of goals (ends) is of prime importance"* (Kroes, 2002, pp 297). The problem arising from this is that there is no apparent continuity between the two contexts with designers unlikely to be end users and vice versa (Kroes, 2002). The lack of continuity between these perspectives was also compounded in the Pronto project by the novel nature of the system, which had little usage precedent, and explicitly sought to distance itself from past practice.

Similarly literature suggests these two different contexts have diverse quality concerns for the system. The developers are concerned with the 'developmental' qualities like maintainability, reusability, etc, while the runtime qualities like performance are of more concern to the user community (Abowd et al., 1997). The desire of the lead designer to introduce some new quality attributes into the ATAM process attest to the different quality concerns associated with the design perspective.

The notion of quality attributes as requiring other qualities suggests a convolution to this relationship where multiple contexts overlap. In the instance of performance and availability, the two properties could be seen to be operating within different contexts but supporting one another in the process. From the user perspective, performance defines the operational expectations that need to be met for them to enjoy the telecommunications service, while the network operations manager will perceive the importance of availability to this. Instead of having separate concerns, these perspectives are better viewed as having different criteria for success. The broad stakeholder group ensures a rich diversity of contexts exist for the system at hand.

> *"...Apart from the context of design and context of use, technical artefacts figure in many other contexts of human action, such as the context of production, context of maintenance, context of consumer markets, etc. Each of these contexts has its own criteria for quality and success, which may be relevant to the way the quality of the design of the artefact is evaluated."* (Kroes, 2002, pp 301)

Hierarchical consistency within systems principles suggest that there is one broader context (outer environment), inside of which are the other contained contexts (Hitchins, 1992, pp 54 & 153). However the reflections discussed how the service subscriber perceives the system as the entire network, whereas the carrier understands the difference in systems delivering the service. Similarly Chapter 5 discussed how these different views of the system at hand manifest themselves in different goals and aspirations of different stakeholder perspectives (Checkland, 2000). The goals of the network administrator cannot be subsumed into that of the broader telecommunications business as they perceive themselves as a separate entity that has relationships to the overall carrier business as well as the customers. Information systems acknowledges this complexity and suggests that models of reality are only *'concepts relevant to explore what we perceive as 'reality''*, developed from a particular world view (W) (Checkland & Holwell, 1998, pp 13). The resulting Soft Systems Methodology seeks to explicitly model the different views of reality from the various perspectives. Promoting the view there is no one reality, just a series of models which have to be understood with respect to their world view.

Such levels of complexity in the ATAM process tend to be masked by the relatively simple attributions that are made in the utility tree. The tree structure and method to derive it doesn't explicitly encourage the use of perspective to make sense of the diversity of stakeholder's perceptions of system quality. This thesis has slowly begun to unravel the true complexity of the seemingly innocuous attributions that are made in the utility tree. To begin with, the business strategic perspective has shown to be an incredibly complex one that is difficult to distil down into dependencies on system quality attributes. The relationship is not an asynchronous one either, with the formulation of the business strategic and systems quality artefacts exhibiting concomitance in Chapter 4 and Chapter 5. Similarly the formulation of the design and the requirements (taken here to include the systems quality perspective) are also known to be interdependent (Curtis et al., 1988).

The behavioural perspective of the system seeks to provide the solid grounds upon which to base understanding (Abowd et al., 1997), yet it relies on the blending of two vastly different domains, that of design (physical structure) and that of use (intentional human action) (Kroes, 2002). Within these domains are individuals that share different views of the system (Checkland & Holwell, 1998), have different object worlds incorporating different language and symbolism (Bucciarelli, 2002), and perceive different goals for the system. Exercises throughout the project such as eliciting definitions, undertaking goal-based requirements and developing glossaries sought to construct the meaning necessary for these stakeholders to communicate. Where the artefacts don't reflect the common understanding, communication begins to break down. For instance the lack of business strategic context in the utility tree caused difficulty early on. Then the failure to use knowledge produced from the GBR artefacts in creating the utility tree forced the group to reconsider its structure.

Communication was also obstructed by the lead designer's use of different terms in trying to convey issues to the broader stakeholder group that had surfaced during design. Consequently, their concerns received little attention from the participants during the sensitivity/trade-off analysis. The silent abdication of responsibility points towards an insidious problem of poor shared understanding begetting poor sense of ownership. Negotiation of meaning as well as objectives proved to be a critical aspect of the first two chapters with large amounts of time spent trying to understand each other's conceptions of goals and qualities. Because the time was available to each of the earlier processes and such understanding was seen as critical, the negotiation of meaning took place. Even, if in all cases the negotiation didn't unambiguously resolve meaning (if that is indeed possible).

In the Pronto ATAM, understanding was hindered by both time pressures and the significant progress that had been made on the design since it was last reviewed. Time pressures tended to push the ATAM steps along. Open discussion concerning meaning took second place to the need to develop the artefact that would mark the end of a stage in the ATAM. This is evident in the comparatively hasty associations made between the systems drivers and quality attributes, as well as the refocusing on the response measure during scenario gathering. The associations were later revised and changed significantly because the participants couldn't clearly recall the basis upon which they were made.

The time since the last review also created a strong need for developing a stakeholder wide understanding of the design. The focus of the earlier ATAM was largely on understanding the business strategic perspective, though learning soon became dependant on systems design issues. During the Pronto ATAM, significant progress had been made on the design and the focus moved to understanding the system architecture with respect to the goals. Honouring the interdependent nature of design and requirements, the lead designer sought to educate the group of consequences they had realised in undertaking the design task. Reflecting on the design moves, designers shift from entertaining possibilities (what if), to understanding the implications of their decisions (Schön, 1991). The difficulty in creating a shared understanding in this instance lies with the detailed design knowledge accrued by the designer in understanding these consequences. It is difficult for them to convey their understanding in the brief and structured time given to them for the architecture presentation. Without constraints to do so, they are also less likely to use the language of the group, as was the case with the lead designer's additions to the utility tree.

The lead designer was also eager to use the ATAM as a forum to resolve some of these implications that directly affected the strategic objectives of the client. However the structure of ATAM is oriented towards the identification of these issues, not their resolution. Perhaps more disappointingly for the designer is the fact that methods of resolution favour the polarity of choice. Whereas reality is often more complex than this and a balance is often sought in these situations, something that is not a very useful response for engineering approaches to problem solving (Checkland, 2000). Figure 6.7 depicts these issues and the existing theoretical framework from Chapter 5.

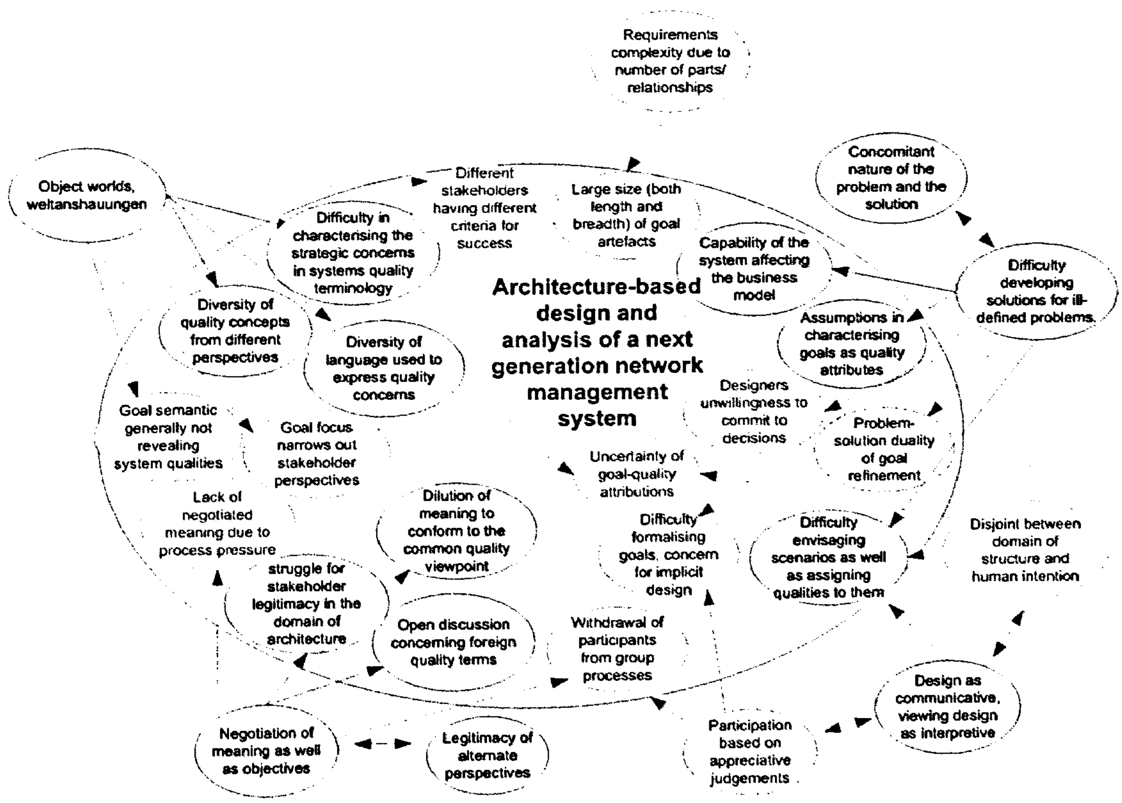

Figure 6.7 - Relationship between the framework and the reflections of the Pronto ATAM

Summary

Chapter 6 reflected on the use of the ATAM to analyse the first cut architecture for the project's NGN management system (codenamed Pronto). The elaboration of the architecture since the previous ATAM activities created the expectation that some of the difficulties arising from inadequate system understanding may diminish. However the reflections testify to the continued difficulties establishing an acceptable utility tree, which forms the backbone of the analysis. The generation of scenarios challenged the group's understanding of the architecture as well as the supporting artefacts, such as quality requirements and driving system needs. The outputs from the quality attribute workshops of the first ATAM and GBR workshops allowed the group to take a well reasoned approach to resolving some concerns, though the group still struggled to develop detailed descriptions of system usage.

Insight into the continued difficulties was offered by viewing technical artefacts as having a dualistic nature. The architecture as presented by the designer exhibits both a structural nature and is also inherently linked to the context of intentional human action. These two facets are disjoint in that the artefact can be analysed for its structural qualities but doesn't naturally reveal the human intent governing its use within a given environment. The lack of shared experience between these domains is confounded by the fact the design and user roles are taken on by different actors in most design situations. The behavioural perspective of the system seeks to provide the solid grounds upon which to base understanding, yet it lies at the intersection of the domains of physical structure and intentional human action. Throughout this project there has been sustained difficulty communicating between these domains with the business strategic perspective proving difficult to distil down into system requirements. A situation made even more problematic by the concomitant nature of relationship. Within these domains are individuals that share different views of the system have different object worlds incorporating different language and symbolism, and perceive different goals for the system.

Overall many of the key themes that had been identified in the previous frameworks were seen to exist within the Pronto ATAM exercise. The learning developed during these earlier exercises allowed understanding to be applied to the problems the group was experiencing and is likely to significantly enhance the ability of the group to analyse architecture in the future. The Pronto ATAM represents the final analysis in this action research project. Chapter 7 will seek to draw an overall coherence to the learning as embodied by the framework and how it has developed across the course of the action research. It will also address learning specific to the methodology of action research and how it has been applied in this project.

7

Learning from the Research and Conclusion

Introduction

This thesis has been concerned with the design of computer-based systems with a focus on the expounding the true complexity of performing architecture-based analysis. The motivation behind the research and its theoretical underpinnings were introduced in chapters 1 and 2. The knowledge aims and research situation were then used to derive the research method of Action Research, presented in chpater3. The remainder of the thesis detailed the action research cycles, reflecting on aspects of applying ATAM (and an associated technique of GBR) within the project. Throughout these action research cycles learning was established with respect to a theoretical framework of ideas created in Chapter 4.

Chapter 7 seeks to bring an overall coherence to the learning outcomes of Chapters 4, 5 and 6 by explicitly addressing their contribution to the research aims. As is evidenced from the key research areas identified in Chapter 1, learning outcomes will concern both the discipline of focus and the discipline of applying Action Research. Consequently the following chapter contains two main sections, the first discussing contributions to architecture-based analysis with respect to the research aims and the second reflecting on the use of the Action Research methodology (Meta-reflections).

The chapter will conclude by discussing the continuation of this research through future work.

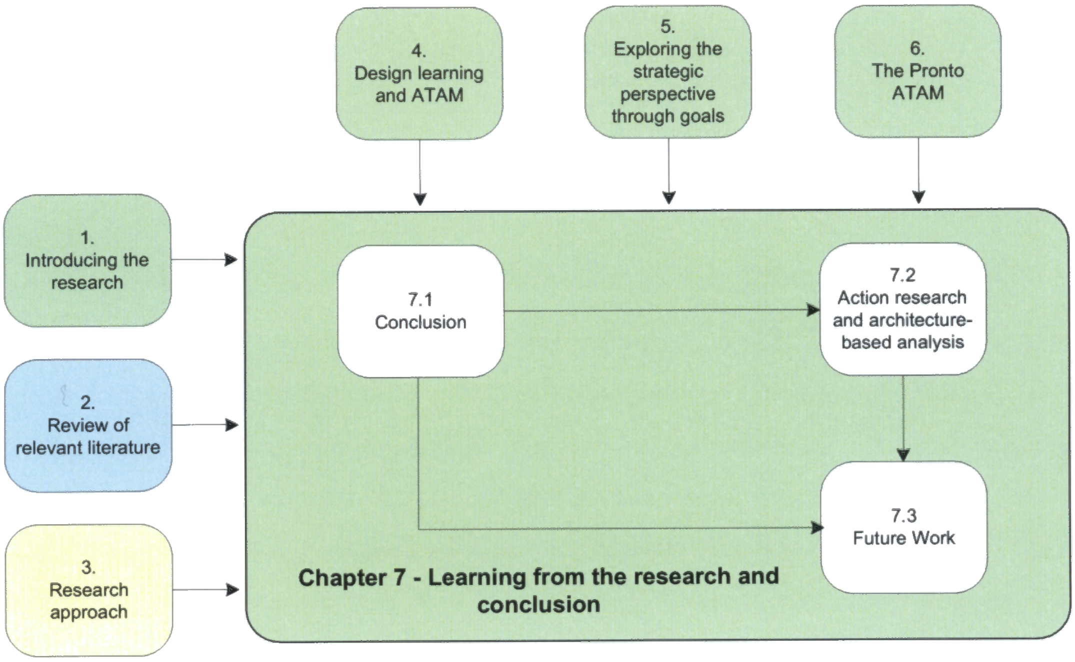

Figure 7.1 - Structure of Chapter 7

7.1. Conclusion

7.1.1. Aspects of Complexity in Architecture-based Analysis

One of the key research aims was to explore the nature of complexity in architecture-based analysis. Drawing upon work by Jackson and Flood, Chapters 1 and 2 put forward the notion that complexity within a situation arises from both the number of parts and relationships of the system at hand, as well as the people within the design situation. These two aspects reveal a dichotomy labelled the 'systems' and 'people' dimensions respectively, as illustrated in Figure 7.2 below.

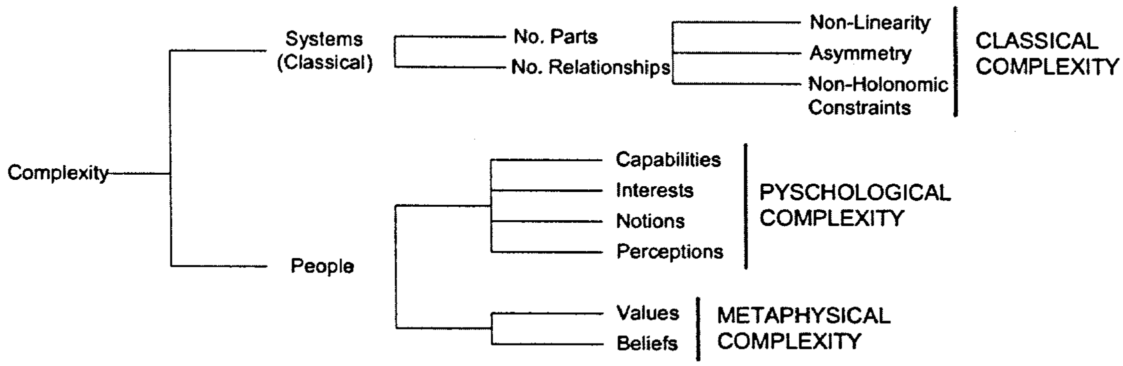

Figure 7.2 - Flood's disassembly of complexity (Flood, 1987)

An examination of the architecture-based analysis literature established that the 'people' dimension to complexity was likely to pervade its processes, yet did not seem to be sufficiently addressed in either its methods or case reporting. Consequently this thesis sought to develop learning on the manifestations of complexity in architecture-based analysis and its subsequent effects on the objectives of the process. The following sections present the aspects of both 'people' and 'systems' complexity encountered within the Architecture Trade-off Analysis Method (ATAM). Section 7.1.2 will then discuss the effect of this complexity on what the method was trying to achieve.

7.1.1.1. People complexity within the ATAM

The people dimension suggests each individual's capabilities, beliefs and interests will manifest themselves in a situation, contributing to the overall situational complexity. Significant evidence for this was found throughout the ATAM exercises, as well as other design activities.

In treating the 'problem of the absent artefact' in design, Galle proposed the conundrum that design representations appear as a misnomer, as the object they represent doesn't exist. His escape from the problem was to acknowledge that representations were representative of the idea and that the stakeholders of the situation make concrete the artefact through their interpretations of it (Galle, 1999). Consequently the act of both creating models and interpreting them has to be understood with respect to the world view (Weltanschauung) of the associated participant(s) (Checkland & Holwell, 1998). Two such world views surfaced early on in the ATAM activities, those of the business strategic perspective and the classic systems quality perspective. The difference between the perspective of the industrial partner and the classical system's quality perspective became evident during the initial presentation activities of the first ATAM. The language of the business drivers as presented by the industrial partner, focused on aspects like cost management and customer satisfaction, which are quite disparate to the generic qualities such as performance and security put forward by the ATAM literature as examples.

The alignment of the greater group with the generic system's quality perspective compared to the more pragmatic business perspective of the industrial partner was also clearly evident in the quality attribute responses. The exercise to elicit the key system quality drivers revealed a similar difference in language and terms between the broader group and the industrial partner. However rather than simply re-enforcing the dichotomy of business and systems quality, the quality responses also indicated a more personal, experience and belief-based view of the quality needs for the system. Stakeholders who were adamant that standards were important to the system offered qualities like 'standards-based', similarly those with broader experience in billing and mobile aspects to networks raised notions like 'roaming' and 'billing accuracy'. Likewise the shortcoming of existing systems presented themselves in the industrial partner's responses such as 'coherency'.

The language used to define the attributes themselves also reflected the personal notions of the participants. Those with backgrounds in telecommunications management used terms such as 'protocols', 'managed data', '5 9's' and 'Frame Loss Rate'. Participants with more hands on networking experience referred to 'rack space', 'moves/adds/changes (MACS)'. Whereas the architecture-minded amongst the group used well known architectural constructs such as 'connections' and 'components' (Perry & Wolf, 1992), the extreme of which was a systems architect who clearly defined 3 qualities then offered the view that these three were contributed to by the remaining qualities.

In situations where such a diversity of viewpoints is evident, the development of artefacts helps the participants to negotiate meaning. The need to create single artefacts representative of the diversity of stakeholder viewpoints creates a tension between the group perspective and the need to satisfy personally held beliefs. Evidence from the goal-based requirements (GBR) exercise suggested that the levels of participation could be contingent upon the clash between the negotiated group artefacts and personally held beliefs. When two group members personally developed goal graphs to extend the work started in the workshops, the broader group became disenfranchised. In such fragmented situations a coalition of two people is effectively a majority and if the speed at which they formulate artefacts is sufficient, the other participants can be left behind (Curtis et al., 1988). Significant change to artefacts developed in a group process without their involvement considerably depletes the group's ability to contribute.

Precedent for this was seen during the quality attributes exercise of the first ATAM where the principal architect took all the quality attribute responses (some 41 in all) and sought to rationalise them down to a workable number and represent it to the group. This sparked several of the participants to air concern about how their responses had been 'interpreted' without their consultation. Conversely where the group has worked hard to establish consensus, negotiated meaning appears largely impermeable to future changes suggested from individual viewpoints. On separate occasions different participants sought to augment the set of quality concerns with a quality they saw as important, following the completion of the quality attribute workshops. While the group could see the importance of these attributes they had not passed through the same process of group negotiation and couldn't be viewed as representative of the group's viewpoint. Similarly when the designer sought to have some personally considered attributes added to the utility tree exercise, rather than being met with open scepticism and questioning, these additions were treated with what was referred to as a silent abdication of responsibility. Although the end result was much the same, the acceptance of the lead designer's attributes in the latter instance also shows the different balances of influence that exist in a project.

This suggests artefacts can't be taken at face value and that participants attach a specific meaning to them. Importantly this meaning is situated and a change of perspective may indeed encourage a different meaning or symbolism as seen in the Pronto utility tree, where the quality-goal associations were seen to change from those decided at the beginning of the process, to those used at the mid-point.

The symbolic meaning attached to artefacts is closely linked the act of appreciation whereby individuals assess reality (what is the case?) and judge value (is this acceptable or unacceptable?). These judgements cannot be ego-less and are intrinsically linked to the participant's view of their role in the situation. On several occasions the lead designer made the point that implicit design assumptions were being made in trying to refined artefacts to a level where scenarios could be specified. Being the designer, any aspects of design that crept into the development of artefacts would directly impact their work and potentially force decisions that they themselves were unwilling to make at that stage. Similarly when the lead architect enlisted the help of a team member to develop the quality attribute groups after the first quality workshop, another team member who saw architecture and hence systems quality as one of their primary concerns, perceived the need to change the quality attributes.

7.1.1.2. Systems dimension to complexity

While the underlying theoretical basis to this research focused on the people dimension to complexity it became clear throughout the action research project that amongst many of the manifestations of 'people' complexity was an equivalent element of 'classical' complexity. Be it due to the intractable nature of design, the notional nature of the system at an early stage of design, or the various ways a system can be decomposed, behind most people problems complex systems aspects could be seen to be co-incident.

One of the most prominent aspects of systems complexity in the ATAM proved to be the relationship between the business strategic and systems requirements perspectives. Early experience in the design meetings served to highlight how intrinsically related the business and technical problems were. Discussion concerning the nature of the design solution would commonly revert back to issues at a business strategic level (where further clarity from the industrial partner was sought). The importance of this relationship was embodied in learning from the first ATAM exercise where significant differences in the candidate architectures could be traced back to subtly different business aims. In the context of systems architecture and architecture-based analysis, these aims are characterised in systems quality terms such as performance, availability, etc. This characterisation has subsequently encouraged the view that customer's needs could be effectively expressed in systems quality terms. However as discovered in the quality attribute workshops of the first ATAM, the business need and the quality attributes are not readily reconcilable. Without significant insight into the business solution assumptions would in all likelihood creep into the systems view.

Acknowledging the importance of the business strategic aspect the group turned to the techniques of goal-based requirements (GBR) in order to obtain requisite insight into them. Goal-graphs were used as the primary mechanism for modelling the business strategic perspective. The GBR literature depicts the goal refinement as providing both the quality (behavioural constraints) and eventually the functional requirements for the system. However quality aspects of the system were only revealed sporadically and implicitly throughout the graphs and detailed refinement did not appear to naturally converge towards either functions or behavioural constraints. The ensuing size of these graphs and the extent to which they could be refined, without revealing significant detail of requirements for the system, availed the design group of the 'depth' behind the strategic perspective. In addition to the notion of depth, goals could also be linked in a contributory sense to any number of other goals, exhibiting extensive 'interconnectedness'. Rather than providing the bridge linking the business perspective to the systems one, the GBR exercise suggested a logical hiatus between them that could not be readily resolved.

The depth and interconnectedness of the goal graphs introduces one of the most traditionally understood dimensions of complexity concerning the number of parts and/or relationships. Miller's 'magical number seven' (Miller, 1956) is reached rather quickly when considering that GBR expects goals to be decomposed in hierarchical relationships, cross-related in contributory relationships, and value judgements placed on the strength of the relationships as well as the importance of the goals. Several goal-graphs of up to eighty nodes were produced, which only represented the higher level considerations (soft goals). Notably, at the soft-goal level the formalism designed to introduce rigour into the goal graphs is highly subjective and is no less likely to produce cascading error as it is incremental evaluation. In the end, the large numbers of goals and relationships were dealt with in a novel and concise way, but by in large the design group found it difficult to work with such numerically complex artefacts during analysis, as witnessed by early attempts to rationalise down the full list of quality concerns.

Although being logically disparate there was no mistaking the fact that the refinement of the goal-graphs held significant consequence for the nature of the design and vice-versa. Here the concomitant nature of the problem and solution came to the fore. In seeking to explore the nature of the solution, learning would inevitably be yielded about the nature of the problem. Neither linkage partner nor the designers would be willing to prematurely commit themselves, hoping to yield the most informed result, yet logically neither the problem nor the solution could be fully developed without extensive knowledge of each other. The concomitance of issues means that the extensive construction of knowledge in one area needs to be conducted with sensitivity to the potential decisions in the other. The sheer number of possibilities is likely to make the task unworkable (Joseph, 1996).

One of the consistent difficulties in attempting to communicate aspects of the solution or problem was the elusive nature of use. Early on in the design lifecycle the system architecture is incomplete, hindering attempts at understanding the potential usage aspects of the system (Brooks, 1987). Complicating matters was the intention that the NGN management framework was to leave behind existing management practises. As such traditions and experience became largely invalid because they were perceived as coming encumbered with the past mistakes. Therefore no real precedent for a Next-Generation Network (NGN) telecommunications system existed and concepts of use which would normally develop from detailed system knowledge or experience were obstructed. In their place abstractions of use, in this case the operational aspects, or operational specifications, were put forward. The problem with operational specifications like those so commonly used in telecommunications (NGOSS, etc) is that they are solution agnostic. They specify what has to be done but don't give clarity on how it should be achieved. A task like the ATAM really requires the behavioural detail behind how things are achieved to understand the quality ramifications. Two systems could quite readily exhibit the same operational characteristics but have two entirely different systems (structurally, architecturally) implementing them.

The previous two sections have discussed the prevalence of complexity in architecture-based analysis, addressing both the people and system dimensions as well as describing the context behind its occurrence. Building on this understanding of complexity and the situations in which it occurred, the following explains the consequences for the ATAM process.

7.1.2. Implications of situational complexity for ATAM the process

Where these aspects of systems and people complexity are coincident upon a process it is understandable they would affect the conduct and outcomes. Perhaps one of the most prominent and enduring of the 'people' aspects to complexity was the difference in perspective between the participants from the linkage partner and the design team members from the university. The linkage viewpoint has been identified with what can be termed the business strategic perspective, typically brought to the design situation by the customer. The design team's viewpoint has been identified with the traditional systems perspective. From the initial difference in quality attribute to the difficulty developing scenarios representative of the business drivers in the Pronto ATAM, the disparate nature of these perspectives was prominent throughout the project. In many ways a majority of the complexity emanates from this conceptual divide, which undermines the very purpose of the ATAM.

> *"The ATAM is designed to elicit the business goals for the system as well as for the architecture. It is also designed to use those goals and stakeholder participation to focus the attention of the evaluators on the portion of the architecture that is central to the achievement of the goals."* (Bass et al., 2003)

The simple relationship between the business goals and the system quality attributes portrayed by the ATAM literature and the subsequent characterisation that takes place belies the complexity of the strategic viewpoint and the complexity of the relationship. Early on the group noted that the language of business drivers and goals were not directly commensurate. Further elaboration of the goals in the GBR exercise showed that quality attributes were implicit in some of the goals but refinement of the goals made no guarantees of yielding explicit quality attributes, akin to those commonly found in software architecture literature. Characterising high level goals as quality requirements, as is the case in the ATAM utility tree potentially suppresses the refinement reasoning required to establish a meaningful relationship between the two. Reasoning that is inevitably rich in appreciative acts of value and reality judgements, made from the perspective of each concerned stakeholder.

The extent and complexity of the strategic viewpoint and the problematic nature of characterising goals as system qualities, calls into question whether the system as judged from the quality viewpoint transitively satisfies the business strategic goals. From the evidence presented in this thesis, this is largely not the case. Although the ATAM addresses important quality concerns, there is no certainty these concerns ensure the satisfaction of the customer's business objectives or allays their greatest fears. The lack of certainty affects the confidence of the group when deriving the attribute utility tree, which is effectively the centrepiece of the analysis. When faced with a utility tree devoid of business context, the stakeholders sought to elaborate the utility tree with aspects of the business drivers. However this soon faltered when the group expressed apprehension as to the rigour of the relationship between the business and systems quality aspects. The group suggested the association between the quality attributes, which the scenarios are derived from and the system business drivers that are most important to the business stakeholder, were effectively ad-hoc. A matrix method was used to help overcome this and while it assisted with the nature of the association, it also highlighted how unconvincing the original relationship was. The group was comfortably able to reason through many changes. The matrix exercise also brought to light the complexity of business drivers, revealing their multiple contexts and how each of these contexts would view the system differently and require different quality aspects to satisfy it.

The difficulties developing the utility tree suggest a lexical and conceptual separation that is not resolved by existing techniques. Neither is its resolution encouraged by the time pressures of the existing process where the need to finalise artefacts in order to progress is clearly evident in the facilitators responsibilities to 'control the pace' and 'maintain authority' (Kazman & Bass, 2002). The facilitator is not an unbiased participant and brings to the process their own world view. Ideally the 'authority' mentioned here would only extend as far as facilitation is concerned. However this thesis has offered evidence to suggest the ATAM takes place in the domain of software architecture, using its language and terminology, on which the facilitator is seen as an authority (Kazman & Bass, 2002). Whether legitimately called upon by the group, like in the styles exercise or assumed as part of their appointment as was the case in manipulating the quality attribute workshop responses, the facilitator will influence proceedings. The facilitator needs to be cognoscente of their effect on both understanding in the process and outcomes. Vickers counsels that in appreciative systems the role of the human regulator in deciding what courses to follow becomes a major influence in the generation of courses (Fischer et al., 1995). Likewise the ATAM literature has already noted the potential mismatch in communications resulting from the business owner and designers having to communicate through an intermediary in the form of a facilitator (Kazman & Bass, 2002). When confronted by the relatively unique quality terms of the linkage partner the facilitator sought to interpret them in more architecturally familiar systems quality terms. The linkage partner's perspective was an important one, yet was unlikely to receive much representation in the process while it differed so significantly from the broader group. The experience reaffirmed the legitimacy of the systems quality perspective and the alignment of the facilitator's world view with it. Instances like this where participant's views are challenged in the face of broader quality frameworks have the potential to ostracise participants from a process where their input is critical.

Conducting the exercise from a specific world view potentially stifles more diverse viewpoints. This thesis has discussed the idea that as well as viewpoints associated with roles, each of the participants occupies their own personal viewpoint that is based on their beliefs and experiences. These beliefs and experiences shape the way in which the participants respond and provides an opportunity for a unique insight which has not yet been appreciated by the rest of the group. Carrying forward Joseph's notion that often design situations benefit most from novel thought that breaks the mould (Joseph, 1996), Chapter 4 presented the situation where participants showed significant interest in each other's qualities when they were tabled. Here we see the importance of Sargent's negotiation of meaning (Sargent, 1994), which should precede any negotiation of objectives or goals, something that is seldom addressed in the ATAM. Although it professes the need for facilitators to seek concurrence and feedback, there is little suggestion as to what this is. Meaning in terms of the ATAM is primarily 'negotiated' through the construction of the utility tree, where ambiguity of meaning is resolved through concrete scenarios. As discussed above, by this stage the intended contribution of the participant may have been significantly altered to conform to quality attribute norms. Similarly scenarios proved no refuge for understanding as early on, the usage context of the system is not well understood and interdependencies with business strategic issues can hamper the development of more detailed design concepts. This represents nothing of the social framework in which Sargent suggests the negotiation of meaning should take place (Sargent, 1994).

By not openly negotiating meaning throughout the process, ATAM tends to leave the participants isolated in their own object worlds, rather than permitting them the ability to evolve their understanding of the situation with respect to their peer's views. This can significantly enhance the analysis process in both the emergent outcomes from constructive understanding as well as the less tangible aspects of providing the group with a common identify and understanding. Operating in 'taken for granted' meaning, rather than group understanding was perhaps no better exemplified by an exercise undertaken between ATAM exercises, in order to define problematic terms. It took several workshops and significant effort before the group was able to agree upon a definition for the word 'service' alone.

It also seems that where group understanding is hard fought it is equally strongly defended. The process does not easily accommodate the entry of new information into proceedings. For example the lead designer's desire to see the concept of 'simplicity' included in the utility tree based on the learning they had accrued in trying to design the system. While the existing quality concerns were established as outcomes from previous group processes associated with the analysis, the simplicity attribute appeared to be perceived as the concern of the person introducing it. Consequently it received little attention during analysis activities. The early artefacts appear to form the bounds of what the group is willing to grant legitimacy to, with negotiated group concepts forming a conceptual equilibrium that is largely inelastic to individual input. Akin to Berger's notions of shared reality construction (Berger 1976 in Truex et al., 1999), what the group perceives as the requirements are the requirements. The cursory inclusion of ideas followed by abdication of responsibility or the outright rejection of new ideas will inherently affect the ATAM where time elapses between activities or the stakeholder group changes. The broadening of the stakeholder group between the first and second phases of the ATAM and the passing of time is likely to experience this resilience of established group opinion over and above that of individual viewpoints.

The inclusion of diverse viewpoints is part of the ATAM's intention to be inclusive of stakeholder groups in the analysis process. Yet there is no real guarantee that participation is equivalent to proper representation. The right to propose ideas and vote on them provides some avenue for granting voice to the stakeholders, yet the structure of the overall exercise and the complexity of multiple world perspectives involved dilutes the outcomes. The ATAM begins with the business goals as the driving elements for deriving the utility tree, yet as discovered in the goal modelling exercise, this is only one perspective on the situation. The goals of the network administrator and customers will differ from those of the business. Indeed the notion of world views suggests they perceive very different systems. Furthermore the 'soft' systems perspective suggests that looking at systems as maintaining relationships rather than achieving goals provides a richer model of the world (Checkland, 2000). Looking at the network management system (including the people within) as being a part of the business, the network administrator goals may be viewed as subsumed into the objectives of the business. However the goal modelling exercise revealed that the network administrator perceived themselves as an entity, which had relationships to the business and the customer. These are best describes as 'holons', which are simultaneously both an autonomous whole and in principle part of larger wholes (Checkland & Holwell, 1998). Neither the ATAM, nor the GBR exercises sought to develop understanding of, and reconcile, such diverse perspective.

7.1.3. Architecture-based Analysis and the Design Lifecycle

Utilising the Architecture-based Design Method as the methodology for investigation provided valuable insight into the iterative application of Architecture-based Analysis within broader design activity. As discussed in Chapter 4, early design activity is focused on exploring the nature of the problem with respect to broader analogies, whilst simultaneously coming to appreciate the uniqueness of the situation at hand. Activity such as this is well explained by Schon's concepts of naming and framing (Schön, 1991). Although ATAM proved a useful framework for design learning, considerations at this stage were seen to be largely driven by uncertainty at the business strategic level, with quality requirements generally representing broader issues within the business drivers.

The complexity of the business strategic viewpoint uncovered by the GBR exercise demonstrated that the strategic solution needs to be as well considered and designed as the systems technical one. Later analysis showed that in addition to the complexity offered by refining and tracing the interdependencies of goals, there are also many external factors directly influencing this perspective. The Pronto ATAM raised the point that aspects presented in the business context concerning the environment of the business provided important rationale behind many of the goals, but was not utilised in any of the artefacts. The pace of technological development and the ability of technology to define market capability constantly change the business environment, forcing companies to continuously redefine themselves creating what Truex has referred to as 'emergent' organisations (Truex et al., 1999). These environmental factors, as well as scarring past experience are largely what the customer brings to the design situation (Joseph, 1996). Likewise the personal perspectives of the design team ensure there is any number of views of the situation early on. At this stage in design the stakeholders are largely conceptually isolated and the focus of early activity is on negotiating meaning.

Once meaning has been negotiated the group can meaningfully discuss the quality aspects of the system in a bid to move from the business strategic to system design considerations. Personal meaning starts to become associated with group beliefs as negotiated artefacts are developed in group situations. However in the Pronto design situation these beliefs were largely symbolic and the real meaning behind them was obfuscated by the lack of clarity surrounding the shape of the system design and its usage context. The Success indicated within existing case reporting has indicated that once the design is of sufficient maturity, usage concepts become embodied with detailed behavioural understanding of the system. The clarity offered by the detailed design aids the group in attributing more specific meaning to terms and reach consensus on meaning. The detail of the design also focuses the group on the technical challenges, testing against what are considered to be reasonably form notions of strategic direction.

Figure 7.3 attempts to depict the progression of the design and the associated characteristics of the ATAM situation. Boehm's spiral model (Boehm, 1988) is used as its overlay to indicate the early stages of design close to the origin and the later stages of design towards to the outer layers.

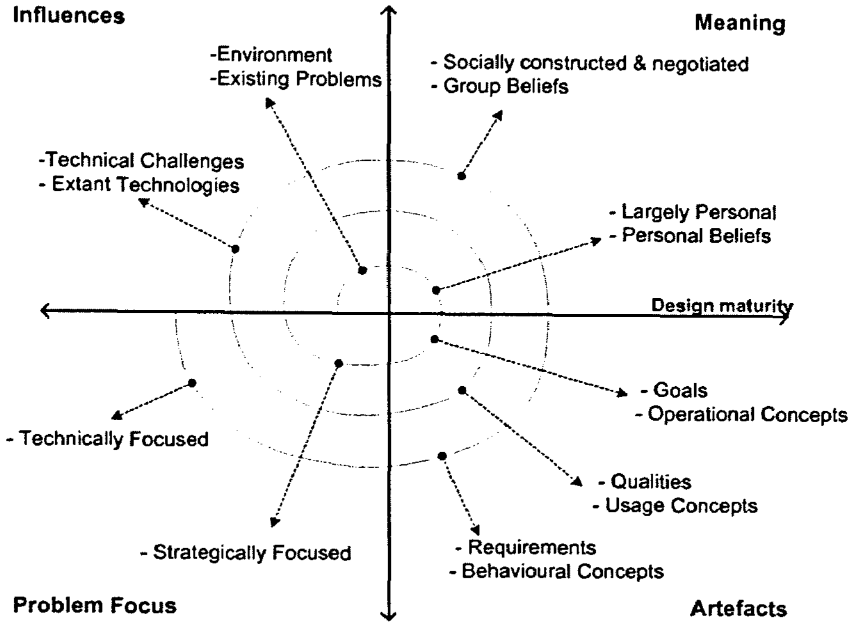

Figure 7.3 - Stages of design and its influence on group process

Evidenced by the experiences throughout this project as depicted in Figure 7.3, the use of ATAM early on is problematic for several reasons. The first is that the ATAM does not focus on making sense of the business strategic viewpoint. Rather it is more focused on using it to develop a list of business drivers. However there are business problems to be solved as much as there are technical ones. Secondly it seeks to move quickly from the business strategic viewpoint into the systems quality one, which is actually a significant evolution in design terms. Lastly, emanating from the quick progression from business perspective to the systems quality one, there is an assumption of group understanding. The difficulty is that this has neither been given the open forums in which it needs to develop, nor the detailed structure of well understood artefacts upon which it is commonly built.

The ATAM relies on well reasoned and stable understanding of the business strategic needs. Problematically these needs are exposed to the constant turmoil of the external environment and tend to develop in concert with, rather than well ahead of the system design. The dependency of the technical solution on business needs means that in order to facilitate such detailed analysis, the strategic benchmarks which the system will analysed against need to be set. The focus at this stage has turned from strategically focused thinking to design activity and significant effort is likely to have been spent. However the co-dependency of the business need on the technical solution suggests revision of the strategic direction upon the basis of understanding at the systems level is both likely and costly. Ideally the concomitant nature of the problem and the solution supports the need to deal with both at the same time.

An approach that could potentially permit analysis at an early stage is to tailor the process according to design stage. Figure 7.3 conveys the idea there are different needs at different times in the design lifecycle. Initially the problem focus is strategic; the strongest influences are previous system failures and changes in the business environment; meaning within the group is largely experience-based and personal; and the system is represented by abstract artefacts such as goals and broad operational concepts.

Experience in the GBR exercise revealed that goals exhibited very similar traits to system qualities, in that they were generally abstract, needed refinement to be properly understood and have interdependencies. Notably the technique used to group goals according to positive interdependencies bears strong similarity with the notion of points of sensitivity in the ATAM, with negative interdependencies at the strategy level representing broad trade-offs. These similarities imply the potential for an ATAM exercise focused on the use of goal-level artefacts to establish the consequences between different strategic approaches. This would yield significant knowledge on business strategic issues permitting the customer stakeholder sufficient, rather than partial representation in the process. It also helps model the customer problems that tend to consistently upset the technical design process. The knowledge from the strategic analysis could then be used to feed into the more technically focused ATAMs once the system has developed significant maturity, which sees the design stakeholders as receiving sufficient representation. The first process is effectively 'for' the customer honouring the social (consensus) paradigmatic approach to design, whereas the second is 'for' the designers honouring the individual expert (fitness for purpose) paradigmatic approach to design (Stumpf, 2002).

While the more strategically focused ATAMs can occur before those of the technically focused ones, this is not meant to suggest they are consecutive, nor that each be used exclusively at the start and end of designing. In looking at the way in which management should be effected in organisations, Weick noted that there are situations to be managed and not problems to be solved (Weick, 1995). Environments are dynamic and the conditions that created the need can always change, unbalancing the resolution. In the same regard, organisations that are strongly defined by their information & communications technology (ICT) capability seek to manage a situation. The environment that gave rise to the need continuously changes (Truex et al., 1999), indeed as it did during this project. Changes in the telecommunications landscape would continuously call into question the strategic and technical approach. Truex suggests this is a natural feature of modern systems design problems and that ambiguity in requirements could not be abstracted back to a core set of stable goals (Truex et al., 1999).

Here we see the importance of Schon's advice that when designing local moves, akin to those technically focused ones must be checked for their coherence against the governing idea (Schön, 1991), in this instance the strategic approach. However as the systems approach may need to alter to accommodate the strategic needs, the strategic approach may need to change in order to accommodate its environment in a sort of double loop learning (Argyris, 1999), where the governing ideas themselves must be questioned.

Here the value of maintaining two streams of analysis becomes apparent. The first stream concerned with the strategic approach and the second with the systems one. While ever the two streams of analysis are bound in the same activity they are likely to negatively impact one another. The strategically focused analysis operates in a world of culturally and personally derived perspectives, where meaning needs to be negotiated, solution concepts are conceptual and the resultant solution is heavily affected by the constant flux in the environment. Although solutions at the systems architecture level are still conceptual, they are often considerably more detailed and take significant time and effort to develop. Thinking at this level is largely the responsibility of the designer(s), communicating their world to the rest of the group through the design and the subsequent behavioural aspects of the system. They bare a dual burden of solving the motivating problem as well as the complex technical challenges that present themselves. Regardless of whether design is viewed as an iterative 'rational problem solving' or 'experiential/reflective' paradigm (Dorst & Dijkhuis, 1995), some degree of stability in the design situation is required, something that is largely unachievable when exposed to the constant change of the strategic perspective (motivating problem).

> *"In extremely complex situations, when uncertainty prevails and when the environment changes rapidly, the optimisation sought by hard systems thinking becomes unobtainable"* (Jackson, 1988, pp 156).

Separating the two types of analysis allows the customer to analyse business issues with the same rigour that systems analysis covers design issues. The outcomes of the business focused analysis would help provide a firm reference from which the quality requirements for the systems analysis can be developed. The business analysis also provides a formal mechanism to confront the constant change in the external environment, rather than continuously exposing the system design to the business uncertainty.

7.1.4. The importance of 'content' versus 'process'

In addition to the need to consider two different but inter-related types of analysis during the design lifecycle, there also needs to be a greater consideration of what analysis activities are achieving. In looking at design methodology Dorst noted that although most methodologies were specified in terms of processes, it was largely the 'content' of the design situation that dictated the designer's actions.

> *"In most cases, considerations linked to the content of the design situation (the perceived design problem, the designer's goals and the perceived possibilities for the next step) will determine the 'kind of action' (process-component)"* (Dorst & Dijkhuis, 1995, pp 265).

It is therefore not surprising that the reality of systematic design conflicts with the way in which it is prescribed (Joseph, 1996). Similarly the ATAM concentrates heavily on the process itself at the expense of the content issues, which can significantly affect an analysis. For example is the goal of the presentation of the business context just that?, or is it to reach agreement and understanding within the group of the strategic issues influencing the system development and to carefully derive a set of system characteristics that can be seen to satisfy the strategic needs? Similarly is the goal of the analysis team in this situation to simply have the information presented? Or is it to create an exhaustive understanding of all the strategic elements, and to realise a rigorously derived set of system characteristics? Not forgetting of course there are multiple perspectives here. Do the designers have few expectations of the business context presentation and participate simply as a passive audience? Or do they expect a comprehensive presentation of issues availed to them through their interaction with the client, which they can rigorously question and seek to understand in a detailed way?

Where the reality of the activities does not fulfil the expectations of a stakeholder, there is little guidance for how resolution may occur. In this project improvisation took the form of glossaries, elicitation of meaning where it was not required, goal-quality matrixes and extensive elicitation of goal artefacts, amongst other things. Although explicit methods were used here to explore the improvisation there is a distinct danger that in commercial settings, with greater time pressures, where the facilitator would feel pressured to maintain 'in control' of the exercise, the improvisation may well take the form of internal judgement and assumption. The danger of which is a self-sealing process (Argyris, 1999), whereby the internal assumptions of the participants are not tested and any incidents that lurch towards difficulties in understanding are avoided lest they hamstring the entire process. However it must be acknowledged that this research can only point to the potential for this to occur due to the experience in the linkage project. The Future work section discusses he need to further examine the protocols used and look at their relationship to the process as specified.

7.2.Action research and architecture-based analysis

Given that the methodology is commonly used in the related field of information systems (IS) but scarcely discussed in relation to the more specific area of systems architecture, the final area of learning focuses on the use of Action Research within the research project. The focus on the application of the research method fulfils the final AR meta-learning requirement of 'process' (Mezirov 1991 in Coghlan & Brannick, 2001). The previous reflections and conclusions can be seen to fulfil the 'content' and 'premises' aspects, although the premises in adopting the AR approach will be discussed.

One of the most prominent affects of conducting Action Research (AR) in the project was the concept of role duality. As the first chapter discussed, AR calls for the researcher to be actively involved in both the project at hand as well as maintaining their own personal AR perspective. These are referred to as the project role and AR role respectively. Depending on how the two roles relate to one another in terms of their flexibility and permeability, role identity can be placed along a spectrum of high role segmentation to high role integration (Ashforth et al 2000 in Coghlan & Brannick, 2001, pp 64 & 80). Permeability refers to the ability for the researcher to be physically undertaking one role and psychologically and/or behaviourally in another, whilst the notion of flexibility refers to the spatial/temporal boundaries (scope) of each role. The research situation outlined in this thesis showed that the roles tended to exhibit good permeability in that design activity was of equal relevance to the collaborative project at hand, as it was the action research project. If not directly concerning analysis itself, the design activity commonly centred on artefacts of importance for the analysis process. While permeability may seem like an ideal situation to have, it also created the difficulty that in the frenzy of project activity it is sometimes difficult to discern in what capacity one is acting. Where high levels of responsiveness are required to overcome situations, on reflection, there was some difficulty discerning whether action was equivalent to being reactive in a project role, or whether it was a lightly planned incursion by the action research project designed to overcome the perceived difficulty.

Conversely to the permeability, the action research project in this thesis demonstrated poor flexibility at times. The Architecture-based Design Method (ABDM) requires the iterative application of architecture-based analysis during the design lifecycle, placing the ATAM research in the critical path of the design effort. The commercial responsibility of the project exposed it to the external pace of change in the next-generation network. Consequently design had to move quickly otherwise it risked the common problem of requirements becoming obsolete within the lifetime of the design project (Truex et al., 1999). Although the largely academic design group were patient with the research aims, at times the researcher experienced distinct difficulty progressing knowledge on an issue at hand, quickly enough to be able to offer academically rigorous input into the project's next actions. Consequently a distinct tension arose between the AR and project roles in terms of timing. The project role acknowledged the urgency and need to progress the design as quickly as possible, though simultaneously the AR role saw distinct deficiencies in the current approach and perceived the need rigorously improve it before moving forward. Coghlan suggests in these situations the action researcher tends to initially align themselves with the organisational (project) role (Coghlan & Brannick, 2001), however in this instance the opposite occurred. The insight into the deficiencies of the existing approach made it easier for the researcher to distance themselves from the design outputs, rationalising that things 'could be done better'. A significant influence on this alignment is likely to be the university environment which tends to equally value the doctoral researcher's needs and the commercial research needs. Alignment is likely to be linked closely with self preservation and in commercial organisations the project imperative for completion commonly prevails over the individuals.

Reacting in an agile way to the needs of the project, termed responsiveness in AR literature, is something that strongly defines Action Research (AR)[10]. One of the key premises in adopting the AR approach was the likely need for change to overcome the difficulties at hand. The various adaptations of the method to enhance group understanding and collect greater depth of meaning during the artefact elicitation as well as the use of other techniques such as GBR and the goal-attribute matrix, all contributed to the resolution of problem situations within the analysis exercises. The acceptance of change as a mechanism for learning within AR granted responsiveness to the research, which may have otherwise struggled to cope with the difficulties encountered. The desire to implement planned change within the situation to overcome these difficulties can be seen to aid the integration of project and research objectives. A more interpretive approach would most likely have realised a greater awkwardness with rather passive observation not seeming to contribute to the resolution of problem situations.

Linked to the notion of observation with a view to understanding versus observation with a view to change and future action is the nature of reflection within AR research projects. Looking back upon the nature of the reflections throughout the length of the project, there is a difference in character between those produced earlier on to those produced towards the end. Awareness of the potential need to identify problems and

[10] http://www.scu.edu.au/schools/gcm/ar/art/arthesis.html

develop methods of overcoming them in keeping with a critical theoretical perspective, the earlier reflections align more closely with the need for diagnosis

These early reflections tended to record events as well as impose initial significance and meaning upon them. In looking at the nature of reflection Argyris noted inconsistencies often exist between the theories used to explain action and the way in which the action itself can be interpreted (Argyris, 1999). These two different types of reflection are termed theory-in-use and espoused theories respectively. The ability for personal beliefs to intrude upon our understanding of a situation is inevitable. It therefore seems prudent to attempt to separate the understanding/interpretation of the reflections and the records of events and situations that gave rise to those interpretations. Splitting reflection and diagnosis provides better grounds for re-appreciating the significance of situations and events at a later stage. Reflecting in this way demonstrates a more mature use of Susman's model of AR, which shows that evaluating the effects of action, specifying learning and diagnosing problems are all separate activities following the 'taking of action' within a situation (Susman & Evered, 1978). The framework of ideas proved invaluable with regard to attempting to maintain some separation between the reflections made during action and the interpretations placed upon them. The persistence of ideas from one cycle to the next and the separation of the observable phenomena from the theoretical explanation assisted greatly in achieving a more mature reflective practice.

7.3.Future Work

Future research in the area of architecture-based analysis is strongly influenced by the learning outcomes of this thesis and the theoretical perspective from which it was undertaken. Chapter 3 described the research approach as a situated method of inquiry that was largely interpretive in perspective but critical by necessity. The consequence of taking such an approach is to trade off in-depth, situated learning with the external validity (generality) of the findings. The research should be understood in the spirit of Schon's 'science before the fact' (Schön, 1991). Consequently future research will focus on furthering the key areas of learning discussed in this chapter.

Understanding the relationship between the design situation and the manifestations of complexity in the ATAM, as well as their consequences for the ATAM outcomes. For example attempting to understand the political environment of organisations undertaking design and looking at how it affects the beliefs and participation of the individuals in the situation.

Examine the use of ATAM throughout an entire design lifecycle, as well as implementing the two streams of analysis referred to above and observing its effects on understanding at the business strategic level and whether it helps to stabilise analysis at the systems technical level.

Interpretive look at analysis conducted by others to see if the process as specified faithfully represents what takes place in the workshops and fulfils the participant expectations.

Linked with looking more closely at the 'content' versus 'process' aspects of the ATAM is the desire to understand the expectations and internal workings of facilitation. For example what the group expects of a facilitator, what they perceive their role as, how these views might conflict, how the facilitator handles conflict and difficult situations where process-guided resolution doesn't seem possible?.

Summary

The ATAM represents a significant evolution for architecture-based analysis techniques. In place of masking the analysis process in the problematic scoring and manipulation of figures with a perceived end of candidate selection, the ATAM has taken responsibility for assisting understanding throughout design. The inclusive nature of the process also ensures that communication amongst the stakeholder community is enhanced. However as discussed in this thesis, the inclusive nature of the process and the conceptual nature of architectures challenges the evolved methods, particularly early on in the conceptual stages of design. The resultant situational complexity impedes some of the key objectives that the ATAM seeks to achieve, such as improved communication and a relationship between systems quality and stakeholder goals.

The impacts on the process can be seen to arise from the structure and perspective of the ATAM and notably extend well beyond the scope of facilitation. The process itself needs to adapt in order to provide the social framework in which the negotiation of meaning and objectives takes place. Diverse viewpoints are more likely to be challenged and smoothed over than to be properly entertained within the existing ATAM process. Left unresolved, poor shared understanding can hinder the completion of tasks and most disturbingly disenfranchise stakeholders. Consequently future research on the ATAM is planned in order to ensure that the process addresses the people challenges as well as the technical challenges of designing.

Bibliography

Abowd, G., Bass, L., Clements, P., Kazman, R., Northrop, L., & Zaremski, A. (1997). *Recommended Best Industrial Practice for Software Architecture Evaluation* (CMU/SEI-96-TR-025). Pittsburgh, Pennsylvania: Sofware Engineering Institute (SEI), Carnegie-Mellon University (CMU).

Amyot, D., & Eberlein, A. (2003). An Evaluation of Scenario Notations and Construction Approaches for Telecommunications Systems Development. *Telecommunications Systems, 24*(1), 61-94.

Antill, L. (1986). The information systems design process: many views of one situation. *Design Studies, 7*(2), 80-86.

Argyris, C. (1999). *On Organizational Learning* (Second ed.). Massachusetts: Blackwell Publishers.

Armbruster, H. (1997). Information infrastructures and multimedia communications: different approaches of the telephone, data, and radio/TV worlds. *Communications Magazine, IEEE, 35*(9), 92-101.

Assi, C., Shami, A., Ali, M. A., Kurtz, R., & Guo, D. (2001). Optical networking and real-time provisioning: an integrated vision for the next-generation Internet. *Network, IEEE, 15*(4), 36-45.

Avison, D., Lau, F., Myers, M., & Nielsen, P. A. (1999). Action Research. *Communications of the ACM, 42*(1), 94-97.

Bachmann, F., Bass, L., Chastek, G., Donohoe, P., & Peruzzi, F. (2000). *The Architecture Based Design Method* (CMU/SEI-2000-TR-001). Pittsburgh: Software Engineering Institute.

Baskerville, R., & Pries-Heje, J. (1999). Grounded action research: a method for understanding IT in practice. *Accounting Management and Information Technologies, 9*, 1-23.

Baskerville, R. L., & Wood-Harper, A. T. (1996). A critical perspective on action research as a method for information systems research. *Journal of Information Techjnology, 11*, 235-246.

Bass, L., Clements, P., & Kazman, R. (1998). *Software architecture in practice* (1st ed.). Boston: Addison-Wesley.

Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice* (2nd ed.). Boston: Addison-Wesley.

Blum, F. H. (1955). Action research - a scientific approach? *Philosophy of Science, 22*(1), 1-7.
Boehm, B. W. (1981). *Software engineering economics.* Englewood Cliffs, N.J.: Prentice-Hall.
Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer, 21*(0018-9162), 61-72.
Boehm, B. W. (1989). *Software risk management.* Washington: IEEE Computer Society Press.
Bosch, J. (2003). Software Architecture - Engineering Quality Attributes. *Systems and Software, 66,* 183-186.
Bosch, J., & Molin, P. (1999). *Software architecture design: evaluation and transformation.* Paper presented at the Engineering of Computer-Based Systems, 1999. Proceedings. ECBS '99. IEEE Conference and Workshop on.
Braa, K., & Vidgen, R. (1999). Interpretation, intervention, and reduction in the organizational laboratory: a framework for in-context information system research. *Accounting Management and Information Technologies, 9,* 25-47.
Brooks, F. P. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *Computer, IEEE, 20*(4), 10-19.
Brooks, F. P. (1995). *The mythical man-month : essays on software engineering* (Anniversary ed. ed.): Addison-Wesley Pub. Co.
Brooks, F. P. (1996). The Computer Scientist as Toolsmith II. *Communications of the ACM, 39*(3), 61-68.
Bubenko, J. (1986). *Information System Methodologies -A Research View.* Paper presented at the IFIP WG 8.1 Working Conference, Noordwijkerhout, the Netherlands.
Bucciarelli, L. (1988). An ethnographic perspective on engineering design. *Design Studies, 9*(3), 159-168.
Bucciarelli, L. (1994). *Designing Engineers.* Mass: MIT Press.
Bucciarelli, L. L. (2002). Between thought and object in engineering design. *Design Studies, 23,* 219-231.
Checkland, P. (2000). Soft Systems Methodology: A thirty year retrospective. *Systems Research and Behavioural Science, 17,* S11-S58.
Checkland, P., & Holwell, S. (1998). *Information, systems, and information systems : making sense of the field.* New York: Wiley.
Checkland, P. B. (1995). Soft Systems Methodology and its relevance to the development of information systems. In F. A. Stowell (Ed.), *Information Systems Provision: The contribution of soft systems methodology.* Maidenhead, UK: McGraw-Hill.
Clements, P., Kazman, R., & Klein, M. (2002). *Evaluating software architectures : methods and case studies.* Boston: Addison-Wesley.
Coghlan, D., & Brannick, T. (2001). *Doing action research in your own organization.* London ;: Sage Publications.
Colquitt, D. (2003). Management Approach Perspective, *Internal Report.* Sydney, Australia: UTS.
Cortese, G., Fiutem, R., Cremonese, P., D'Antonio, S., Esposito, M., Romano, S. P., & Diaconescu, A. (2003, January 2003). AQUILA: Adaptive Resource Control for QoS Using an IP-Based Layered Architecture. *IEEE Communications,* 54-60.

Crotty, M. (1998). *The foundations of social research : meaning and perspective in the research process*. St Leonards, NSW: Allen & Unwin.
Curtis, B., Krasner, H., & Iscoe, B. (1988). A field study of the software design process for large systems. *Communications of the ACM, 31*(11), 1268-1287.
Dardenne, A., Fickas, S., & van Lamsweerde, A. (1991). *Goal-directed concept acquisition in requirements elicitation*. Paper presented at the Software Specification and Design, 1991., Proceedings of the Sixth International Workshop on.
Dardenne, A., Lamsweerde, A. v., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming, 20*, 3-50.
Davies, L. J. (1988). How SSM deals with complexity. *Transactions on Instrumentation Measurement and Control, 10*(3), 130-138.
Davis, A. M. (2005). *Just enough requirements management : where software development meets marketing*. New York: Dorset House.
Dick, R. (1993). *You want to do an action research thesis? : how to conduct and report action research, (including a beginner's guide to the literature)* (v2.06;930507. ed.). Queensland: Interchange.
Dobrica, L., & Niemela, E. (2002). A survey on software architecture analysis methods. *Software Engineering, IEEE Transactions on, 28*(7), 638-653.
Dorst, K., & Dijkhuis, J. (1995). Comparing paradigms for describing design activity. *Design Studies, 16*(2), 261-274.
Engel, T., Granzer, H., Koch, B. F., Winter, M., Sampatakos, P., Venieris, I. S., Hussmann, H., Ricciato, F., & Salsano, S. (2003, January 2003). AQUILA: Adaptive Resource Control for QoS Using an IP-Based Layered Architecture. *IEEE Communications*, 46-53.
Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine.
Fischer, G., Redmiles, D., & Williams, L. (1995). Beyond Object Oriented Technology: Where Current Approaches Fall Short. *Human-Computer Interaction, 10*, 79-119.
Flegkas, P., Trimintzios, P., & Pavlou, G. (2002). A policy-based quality of service management system for IP DiffServ networks. *Network, IEEE, 16*(2), 50-56.
Flood, R. L. (1987). Complexity: a Definition by Construction of a Conceptual Framework`. *Systems Research, 4*(3), 177-185.
Flood, R. L. (1988). Situational complexity, systems modelling and methodology. *Transactions on Instrumentation Measurement and Control, 10*(3), 122-129.
Galle, P. (1999). Design as an intentional action: a conceptual analysis. *Design Studies, 20*, 57-81.
Glass, R., Ramesh, V., & Vessey, I. (2004). An Analysis of Research in Computing Disciplines. *Communications of the ACM, 47*(6), 89-94.
Glass, R. L. (1994). The software-research crisis. *Software, IEEE, 11*(6), 42-47.
Glass, R. L. (1995). A Structure-Based Critique of Contemporary Computing Research. *Journal of Systems and Software, 28*(1), 3-7.
Gurp, J. v., & Bosch, J. (2000, May 2000). *Automating Software Architecture Assessment*. Paper presented at the Subworkshop on extreme programming (NWPER2000), Lillehammer.
Hilliard, R., Kurland, M., Litvintchouk, S., Rice, T., & Schwarm, S. (1996). *Architecture Quality Assessment, version 2.0*: MITRE.

Hitchins, D. K. (1992). *Putting systems to work.* Chichester ; New York: Wiley.
Holbrook, H. (1990). A scenario-based methodology for conducting requirements elicitation. *Software Engineering Notes (ACM SIGSOFT), 15.*
Hope, M., & Nichols, E. (2002). *OSS: New Approaches for Successful Service Management*: OVUM.
Houkes, W. (2002). Design and use as plans: an action-theoretical account. *Design Studies, 23,* 303-320.
IEEE. (1998a). IEEE guide for developing system requirements specifications, *IEEE Std 1233, 1998 Edition.*
IEEE. (1998b). IEEE recommended practice for software requirements specifications, *IEEE Std 830-1998.*
IEEE. (1998c). IEEE Standard for a Software Quality Metrics Methodology, *IEEE Std 1061-1998.*
IEEE. (1999). IEEE standard for application and management of the systems engineering process, *IEEE Std 1220-1998.*
IEEE. (2000). IEEE Recommended practice for architectural description of software-intensive systems, *IEEE Std 1471-2000* (pp. i-23).
ITU-T. Telecommunications Management Network, *M.3000 Series*: ITU Telecommunication Standardization Sector.
Jackson, M. C. (1988). Systems methodologies as complementary tools for managing situational complexity. *Transactions on Instrumentation Measurement and Control, 10*(3), 155-160.
Jackson, M. J. (1995). *Software requirements & specifications : a lexicon of practice, principles, and prejudices.* ; Wokingham, England: Addison-Wesley Pub. Co.
Jagodzinski, P., Reid, F. J. M., Culverhouse, P., Parsons, R., & Phillips, I. (2000). A Study of electronics engineering design teams. *Design Studies, 21,* 375-402.
Jamal, T., & Eyre, M. (2003). Legitimation Struggles in National Park Spaces: The Banff Bow Valley Round Table. *Journal of Environment Planning and Management, 46*(3), 417-441.
Janes, F. R. (1988). interpretive structural modelling: a methodology for structuring complex issues. *Transactions on Instrumentation Measurement and Control, 10*(3), 145-154.
Joseph, S. (1996). Design systems and paradigms. *Design Studies, 17,* 227-339.
Kazman, R., Abowd, G., Bass, L., & Clements, P. (1996). Scenario-based analysis of software architecture. *Software, IEEE, 13*(6), 47-55.
Kazman, R., Asundi, J., & Klein, M. (2002). *Making Architecture Design Decisions: An Economic Approach* (CMU/SEI-2002-TR-035)). Pittsburg: Software Engineering Institute (SEI).
Kazman, R., Barbacci, M., Klein, M., Jeromy Carriere, S., & Woods, S. G. (1999). *Experience with performing architecture tradeoff analysis.* Paper presented at the Software Engineering, 1999. Proceedings of the 1999 International Conference on.
Kazman, R., & Bass, L. (2002). Making architecture reviews work in the real world. *Software, IEEE, 19*(1), 67-73.

Kazman, R., Bass, L., Abowd, G., & Webb, M. (1994). *SAAM: a method for analyzing the properties of software architectures.* Paper presented at the Software Engineering, 1994. Proceedings. ICSE-16., 16th International Conference on.

Kazman, R., Klein, M., & Clements, P. (2000). *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004). Pittsburgh: Software Engineering Institute.

Keinonen, T. (2001). *User experience with communication technology.* Paper presented at the Telework.

Klein, H. K., & Myers, M. D. (1999). A set of principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly, 23*(1), 67-94.

Klein, M., & Kazman, R. (1999). *Attribute-Based Architecture Styles* (CMU/SEI-99-TRR-022). Pittsburgh: Software Engineering Institute.

Kroes, P. (2002). Design methodology and the nature of technical artefacts. *Design Studies, 23*(3), 287-302.

Kruchten, P. B. (1995). The 4+1 View Model of architecture. *IEEE Software, 12*(0740-7459), 42-50.

Lamsweerde, A. v. (2000). *Requirements engineering in the year 00: a research perspective.* Paper presented at the International Conference on Software Engineering, Limerick, Ireland.

Lamsweerde, A. v. (2001). *Goal-oriented requirements engineering: a guided tour.* Paper presented at the Fifth IEEE International Symposium on Requirements Engineering, 2001.

Lamsweerde, A. v. (2003). From Systems Goals to Software Architecture. In P. Inverardi (Ed.), *Formal Methods for Software Architectures* (pp. 25-43): Springer-Verlag.

Land, R. (2002). *A Brief Survey of Software Architecture* (Report). Vasteras: Department of Computer Engineering, Mälardalen University.

Lee, J., & Hsu, K.-H. (2002). Modeling software architectures with goals in virtual university environment. *Information and Software Technology, 44*(6), 361-380.

Lee, J., Xue, N.-L., & Kuo, J.-Y. (2001). Structuring requirements specifications with goals. *Information and Software Technology, 43*, 121-135.

Letier, E., & Lamsweerde, A. v. (2002). *Agent-based tactics for goal-oriented requirements elaboration.* Paper presented at the 24th International Conference on Software Engineering, Orlando, Florida.

Lindberg, K. R. (1999). Software developer perceptions about software project failure: a case study. *The Journal of Systems and Software, 49*(2-3), 177-192.

Liu, L., & Yu, E. (2004). Designing information systems in a social context: a goal and scenario modelling approach. *Information Systems Journal, 29*, 187-203.

Maher, M. L., Poon, J., & Boulanger, S. (1996). Formalising design exploration as co-evolution: a combined gene approach. In J. S. Gero (Ed.), *Advances in formal design methods for CAD.* London, UK: Chapman and Hall.

Main, M., Hall, P., Dawson, J., & Stevenson, I. (2002). DSL: Strategies for Next-Generation Services: OVUM.

McCall, J. A., Richards, P. K., & Walters, G. F. (1977). Factors in Software Quality, RADC TR-77-369, *US Rome Air Development Center Reports* (Vol. I, II, III). NTIS.

Medvidovic, N., & Taylor, R. N. (2000). A classification and comparison framework for software architecture description languages. *Software Engineering, IEEE Transactions on, 26*(1), 70-93.
Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review, 63*, 81-97.
Morrison, J., & George, J. F. (1995, July 1995). Exploring the software engineering component in MIS research. *Communications of the ACM, 38*, 80-91.
Mylopoulos, J., Chung, L., & Nixon, B. (1992). Representing and using nonfunctional requirements: a process-oriented approach. *Software Engineering, IEEE Transactions on, 18*(6), 483-497.
Obbink, H., Kruchten, P., Kozaczyynski, W., Postema, H., Ran, A., Dominick, L., Kazman, R., Tracz, W., & Kahane, E. (2002). *Software Architecture Review and Assessment (SARA) Report*: SARA Working Group.
Pahl, G., & Beitz, W. (1984). *Engineering design* (K. Wallace, Trans.). London: Design Council.
Perry, D. E., & Wolf, A. L. (1992). Foundations for the Study of Software Architecture. *SIGSOFT Softw. Eng. Notes, 17*(4), 40-52.
Rechtin, E. (1991). *Systems Architecting: creating and building complex systems*. Englewood Cliffs, NJ: Prentice Hall.
Redwine, S. T., & Riddle, W. E. (1985). *Software Technology Maturation*. Paper presented at the 8th International Conference on Software Engineering.
Regev, G. (2003). *A Systematic Paradigm for Early IT System Requirements Based on Regulation Principles: The Lightwitch Approach*. Ecole Polytechnique Federale de Lausanne, Lausanne.
Rolland, C., Souveyet, C., & Achour, C. B. (1998). Guiding goal modeling using scenarios. *Software Engineering, IEEE Transactions on, 24*(12), 1055-1071.
Roozenburg, N. F. M., & Cross, N. G. (1991). Models of the design process: integrating accross the disciplines. *Design Studies, 12*(4), 215-220.
Ross, D. T., & K. E. Schoman, J. (1979). Structured analysis for requirements definition, *Classics in software engineering* (pp. 363-386): Yourden Press.
Sankaran, S. (2001). *Effective change management using action learning and action research : concepts, frameworks, processes, applications*. Lismore, N.S.W.: Southern Cross University Press.
Sargent, P. (1994). Design science or nonscience. *Design Studies, 15*(4), 389-402.
Schön, D. A. (1991). *The reflective practitioner : how professionals think in action*. Aldershot, England: Arena.
Seaman, C. B. (1999). Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering, 25*(4), 557-571.
Shaw, M. (2001). *The coming-of-age of software architecture research*. Paper presented at the Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on.
Shaw, M., & Garlan, D. (1996). *Software architecture : pespectives on an emerging discipline*. Upper Saddle River, N.J.: Prentice Hall.
Sheridan-Smith, N., Soliman, J., & Colquitt, D. (2004). *Improving the user experience through adaptive and dynamic service management*. Paper presented at the

Australian Telecommunications Networks and Applications Conference, Bondi Beach, Sydney.
Sheriden-Smith, N., Colquitt, D., & Wootton, J. (2003). Moving from Next-Generation Networks to Enriched-Experience Networks, *Internal Report* (pp. 139). Sydney, Australia.
Stevens, M. L., & Weiss, W. J. (1999). Policy-based Management for IP Networks. *Bell Labs Technical Journal*, 75-94.
Stumpf, S. C. (2002). Talking about team framing: using argumentation to analyse and support experiential learning in early design episodes. *Design Studies, 23*(1), 5-23.
Susman, G. I., & Evered, R. D. (1978). An Assessment of the Scientific Merits of Action Research. *Administrative Science Quarterly, 23*, 582-603.
The-Standish-Group. (1994). The CHAOS Report. Boston, Massachusetts: The Standish Group.
Tichy, W. F. (1998). Should computer scientists experiment more? *Computer, 31*(5), 32-40.
TMF. (2000). Telecommunications Operations Map, *GB910 2000*: Telecommunications Management Forum.
Triminitzios, P., Andrikopoulos, I., Pavlou, G., Flegkas, P., Griffin, D., Georgatsos, P., Goderis, D., T'Joens, Y., Georgiadis, L., Jacquenet, C., & Egan, R. (2001). A Management and Control Architecture for Providing IP Differentiated Services in MPLS-Based Networks. *IEEE Communications*, 80-88.
Truex, D., Baskerville, R., & Klein, H. (1999). Growing systems in emergent organisations. *Communications of the ACM, 42*(8), 117-123.
Uglow, S. (2002). *Structuring for survival: Strategies for Telecoms players* (1 904173-09-8): OVUM.
van Gurp, J. B., J. (2000). *SAABNet: Managing qualitative knowledge in software architecture assessment VO -*. Paper presented at the Engineering of Computer Based Systems, 2000. (ECBS 2000) Proceedings. Seventh IEEE International Conference and Workshopon the.
van Lamsweerde, A., & Letier, E. (2000). Handling obstacles in goal-oriented requirements engineering. *Software Engineering, IEEE Transactions on, 26*(10), 978-1005.
Vandermeulen, F., Vermeulen, B., Demeester, P., Steegmans, F., & Vermeulen, S. (2002). A generic architecture for management and control of end-to-end quality of service over multiple domains. *Computer Communications, 25*, 148-168.
Vessey, I., Ramesh, V., & Glass, R. L. (2005). A unified classification system for research in the computing disciplines. *Information and Software Technology, 47*, 245-255.
Vidgen, R. (1996). *A Multiple Perspective Approach to Information System Quality*. University of Salford.
Vidgen, R. (1997). Stakeholders, soft systems and technology: separation and mediation in the analysis of information system requirements. *Information Systems Journal, 7*, 21-46.
Weick, K. (1995). *Sensemaking in Organizations*: Thousand Oaks : Sage Publications.

West, D., & Stansfield, M. H. (2001). Structuring Action and Reflection in Information Systems Action Research Studies Using Checkland's FMA Model. *Systemic Practice and Action Research, 14*(3), 251-281.
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.
Zelkowitz, M. V., & Wallace, D. (1997). Experimental validation in software engineering. *Information and Software Technology, 39,* 735-743.

Appendix A

The case for understanding social complexity in the architecture-based analysis process

David Colquitt[1]
John Leaney[2]
Tim O'Neill[1]

[1]Faculty of Engineering
[2]Faculty of Information Technology
University of Technology Sydney
Sydney, New South Wales
Email: david.colquitt@uts.edu.au

Abstract

Systems architecture is a discipline that seeks to model the abstract form of a system and reason about the qualities of the end system artefact with respect to the design representation. The analysis need has driven the development of several architecture-based evaluation techniques, which have evolved over the past decade from expert-centric, to stakeholder-centric analysis. The resulting group of participants can be considered, as they are in the broader design process, a human activity system, granting architecture-based analysis many of the attributes of a social or 'soft' process. The following paper examines the development of architecture-based evaluation techniques in light of soft systems theory and makes the case for the existence of, and need to understand, social complexity within the analysis process.

INTRODUCTION

The intersection of the technically oriented domain of Information Technology (IT) and its organisational counterpart Information Systems (IS) has broadened the way in which we think about systems and importantly how we approach their development. Systems design lies firmly at the intersection of the IS and IT perspectives where the desire to express the true complexity of the system and its organisational context has to be balanced against the need for a prescriptive statement of requirements from which a technological solution can be derived (Checkland & Holwell, 1998).

The need to understand and balance the antagonistic forces of IT and IS is evident in the commonplace modelling of information systems, several layers of abstraction from the technological system itself (Zachman, 1984) as well as the more inclusive attitudes towards the design stakeholder group (Bucciarelli, 2002). However broader participation and modelling alone is insufficient in representing the IS perspective when the premise upon which the system is modelled originates from a 'hard' systems perspective. Checkland refutes the notion that organisations are simply goal seeking entities utilising information systems in support of decision making, targeted at achieving those goals (Checkland & Holwell, 1998). His soft systems methodology (SSM) promotes the way in which IS models are derived and interpreted as being key to handling the true complexity of systems.

With the organisational context and business goals providing the basic drivers to which all system requirements should be traceable (van Lamsweerde, 2001), the evolved IS perspective clearly holds consequence for any technique seeking to reconcile system capability against the business context giving rise to it, as is the case for architecture-based evaluation.

Arising within the IT discipline, architecture-based evaluation has focused on the need for methods that deal with the technical issues of system development at the expense of more ill-structure problem elements. As is the case for the broader design discipline (Bucciarelli, 2002), research within this field has been instrumental in nature, oriented towards declaring and applying method extensions. By acknowledging the case for social complexity as an issue within architecture-based evaluation we seek to the lay the platform for and legitimate a qualitative approach to researching the application of architecture-based evaluation and how this can be used to improve the efficacy of the method. The following paper discusses the influences that have led to the shift towards systems architecture as a viable approach to complex systems problems and how this has subsequently imposed upon the way systems architecture analysis is conducted. It then discusses the consequences of these changes with respect to soft-systems thinking and proposes techniques to deal with the additional dimensions of complexity.

DESIGN AND SYSTEMS ARCHITECTURE

The necessity of early stage design reasoning

A natural part of the design process is the making of decisions, both consciously through deliberate design choices and implicitly through commitment to the decision and acceptance of its consequences (Schön, 1991). The consequences alluded to by Schön have many dimensions, two of which are the new issues raised in realising the decision and the pruning of previous options from the theoretical "decision tree ", referred to by Simon as a hierarchy of interdependent sub-solutions (Simon1992 in (Joseph, 1996)) . The notion of deepening commitment is further supported by some fairly cogent arguments about the cost of late correction of requirements errors within a partially or fully developed system, when compared with the cost of early correction within the requirements engineering process (Boehm, 1981).

A logical inference from these observations is the further along the system design and development path, the more committed you are to the solution and the more costly it becomes to change earlier design decisions (Abowd et al., 1997). The flow on effect of design decisions influencing each other and subsequently affecting the set of possible solutions, coupled with the cost of "late" changes, raises the profile of the earliest design decisions as being critical to the efficacy of the end solution. The situation is analogous to the built environment where the underlying structure (architecture) contributes most significantly to the properties of the end-product (construction). The relationship between the two fields has led to the popularisation of the term software architecture (Rikard Land, 2002) and in the spirit of 'no software is an island', systems architecture.

The case for architecture

Systems architecture is intended to facilitate designers being able to reason about the structure of a system in an abstract form, free from the constraints of implementation detail. Far from being the single amorphous collection of shapes and lines, depicting a solitary aspect of the system (typically function), so commonly found in papers today, architecture is a very rich concept. A key Neo-Platonic tenet of architecture is that systems are represented by collections of views. Each view representing a certain aspect of the system, for example a functional view to depict key system functions and data flow at varying layers of abstraction (Kazman & Bass, 2002); a process view for synchronisation and concurrency (Kruchten, 1995) and a physical view for showing the mapping of software onto hardware (Bass et al., 1998; Kruchten, 1995). Naturally there are as many "views" of a system as there are logical ways of partitioning and reasoning about a system. Views are represented by one or more models in adherence to the logical separation of the perspective and the representation (IEEE, 2000). In turn models are comprised of components and connections, each with specific properties allowing designers to reason about the 3 key dimensions of a system; data, function and behaviour (Budgen, 1994).

The generic building blocks for creating architectural representations provide a powerful and flexible way for reasoning about systems in general. Significantly it has been highly influential in furthering knowledge about non-functional aspects of systems design. Early development methods for computer-based systems were dominated by the need to capture, describe, grant, allocate and verify function. However it soon became apparent whilst function was important it was generally the non-functional (quality) aspects of systems that caused them to be perceived as failures (Bass et al., 1998). In a world where computer-based systems are pervading most aspects of society and growing increasingly more complex in the process, issues of performance, reliability, availability, maintainability, security, etc, have become of equal if not more importance than function.

Early work by Parnas (Parnas, 1972) showing the application of information hiding principles in system decomposition to grant greater flexibility and Stevens, Myers and Constantine's (Stevens et al., 1974) work with coupling and cohesion laid the groundwork for the creation and use of predictive measures of software quality (Kazman et al., 1994). Attention has since turned from predictive measures and quality metrics towards ready-made design solutions in the form of architectural styles and design patterns.

"An architectural style is a description of the component types and a pattern of their run-time control and/or data transfer. A style can be thought of as a set of constraints on an architecture - constraints on component types and their interactions –and these constraints define a set or family of architectures that satisfy them" (Bass et al., 1998)

The Software Engineering Institute (SEI) at Carnegie-Mellon University (CMU) championed the cause of architecture styles through the work of Mary Shaw and David Garlan. In observing the abstract form of software systems, they made the observation that coarse grained patterns of interaction tended to repeat themselves throughout different systems, as did component types and their generic functions. The importance of these styles to the broader software architecture and design communities was the fact these styles were commonly aimed at providing for some desired system quality, such as performance, robustness, etc. This provided a crucial causal link between the essential structure of a system and the quality attribute goals.

In further developing the notion of "styles" to more concrete instantiations within software, the notion of design patterns were presented . Design patterns (Gamma et al., 1995) are prescribed configurations of objects in response to a problem context, with the aim of granting specified functional and non-functional properties. The importance of design patterns to the budding system architecture community was the causal relationship between the structure of the objects and the resulting quality goals.

Similar to architectural styles, design patterns present configurations of components with specific properties as being capable of satisfying quality design goals. However dissimilar to design patterns, architectural styles are perhaps the earliest design decisions made, dealing with the abstract arrangement of components and connections, rather than the more concrete notions of software objects.

The knowledge that now exists about quality attributes, how they can be identified, measured and realised, has posited a highly significant relationship between the goals of a system, be they functional or non-functional and the early design structure of the system. Importantly it promotes the capability to design for performance, for maintainability or for security through architecture-based decisions.

Inherent in the need to design for particular qualities is the need to test for them. As witnessed by the testing phases of traditional software engineering approaches (Boehm, 1988; Pressman, 1996) and the various validation stages within the systems engineering approaches (IEEE, 1999; ISO/IEC, 2002), all processes that seek to guide the development of systems need to incorporate rigorous elements to ensure the intended outcomes. Without the ability to evaluate architecture-based design decisions against the quality attributes, architecture offers little benefit over existing methods of engineering as shortcomings in the design will remain undiscovered until the later stages of implementation, incurring the same costly penalties alluded to by Boehm.

ARCHITECTURE-BASED EVALUATION

The main purpose of existing architecture-based analysis techniques is to assess the extent to which quality concerns have been addressed in the system architecture and the risk associated with the design (Dobrica & Niemela, 2002). In terms of architecture-evaluation, risks are identified as important architecture decisions which haven't yet been made and hold significant consequence for a particular design goal (Kazman et al., 2000).

In a report on "Recommended Best Industrial Practice for Software Architecture Evaluation" Abowd identified two main types of approach to architecture analysis, *questioning* and *measuring* (Abowd et al., 1997). Questioning techniques incorporated the use of scenarios, checklists and questionnaires, whereas measuring techniques incorporated the use of modelling and simulation as well as metrics. In general the report remarked that measuring techniques were good for exploring specific issues such as performance, but were limited in their generality and suffered from higher resource needs for activities such as prototype development. Conversely questioning techniques offered less rigorous investigation of issues (Kazman et al., 1999) but are capable of being applied to explore multiple quality attributes without the need for development of complex models or simulations. Although the report did not commit to any specific technique as ideal it favoured the use of scenarios, an attitude that has persisted through all of the subsequent SEI analysis initiatives and most of the other existing techniques with only formal code metrics being integrated into the SAABNet analysis framework.

Existing analysis methods

Architecture-based analysis techniques clearly have an important part to play if architecture-based design principles are to reach critical mass, however its significance has not been reflected in terms of interest within the research or commercial community. The lack of attention was noted by Kazman back in 1994 (Kazman et al., 1994) and recent surveys suggest that not a great deal of exposure has been gained since and architecture evaluation still persists only in research circles (Dobrica & Niemela, 2002). Further to this, of the literature reviewed only permutations of 2 of the 6 methods discovered have been reviewed or reported in case-study developments, those of ATAM and its predecessor SAAM (Dobrica & Niemela, 2002; Rikard Land, 2002; R. Land, 2002; Lopez, 2003).

Quality Function Deployment (QFD) (Hauser & Clausing, 1988)	1988
Rank Matrix Analysis (RMA) (Hitchins, 1992)	1992
Software Architecture Analysis Method (SAAM) (Kazman et al., 1994)	1994
Quantified Design Space (QDS) (Shaw & Garlan, 1996)	1996
Architecture Quality Assessment (AQA) (Hilliard et al., 1996)	1996
Architecture Trade-off Analysis Method (ATAM) (Kazman et al., 1999)	1999
Architecture-Level Modifiability Analysis (ALMA) (Bengtsson et al., 2000)	2000
Software Architecture Assessment using Bayesian Networks (SAABNet) (van Gurp, 2000)	2000
Software Architecture Requirements Assessment (SARA) (Obbink et al., 2002)	2002

Table 1 - Existing Published Architecture-based Evaluation Techniques

From evaluation to analysis

The chronology and orientation of the analysis methods presented in Table 1 above, shows two distinct periods in which architecture analysis methods were actively researched and proposed. The first period (1988 – 1996) was marked by the development of QFD, RMA, SAAM, QDS and AQA, when notably all the methods were "questioning" in nature and incorporated the use of numeric values and weightings. The use the matrix-based evaluation frameworks in the earliest methods of RMA and QFD appear to have had a significant influence on the subsequent methods of SAAM, QDS and AQA. These methods were very much evaluation oriented in that they provided input requirements and design configurations as unquestionable statements of system purpose and structure, and then sought to score and select specific design approaches that best suited the requirements. Apart from the apparent difficulties in reliably scoring system designs (Hitchins, 1992) there was also a lack of emphasis on understanding the interdependencies within sub-systems, design approaches and quality attributes as opposed to just the relation between them. Ultimately these methods provided a way of selecting design approaches but allowed no further learning as to how to improve the end solution, in order to account for any inconsistencies encountered during evaluation. They addressed few of the concerns raised in the opening paragraphs about needing to understand and reason about the earliest design decisions in order to prevent costly changes late in the system life-cycle (Houkes, 2002). They were effective selection tools but not effective design learning tools.

Consequently the second epoch of architecture evaluation (1999-2002) witnessed a shift in both technique method and purpose with the publication of ATAM. While paying homage to its predecessor SAAM for the scenario-based evaluation modus operandi, ATAM distanced itself from the numeric assignment of values to capability by declaring a focus on architectural risk. ATAM worked from the understanding that the perfect system was unattainable and in reality designing was the act of managing the trade-offs between conflicting quality requirements in a way that allowed the stakeholder to achieve their business goals. Instead of simply selecting amongst candidate design options ATAM promoted the development of customer goals, the association of these goals to the system quality drivers, the documentation of design strategies to fulfil these drivers and the identification of points in the architecture where multiple quality attribute concerns intersected. By identifying aspects of the design that required greater care when designing and fostering further understanding of both the requirements and design approach, methods like ATAM and SARA have evolved to fulfil not just an evaluation, but an effective analysis role.

From expert-centric to stakeholder centric, expanding the stakeholder group

The progression of architecture-based analysis techniques towards fulfilling a design analysis role has been accompanied by the widening of process scope from involving a few technical experts to taking on a broader role of uniting the stakeholder community, in accordance with the "democratisation of the design process" (Joseph, 1996). A stakeholder community that inevitably grows in reaction to realisations about the implications of the business and its people upon the systems they use, a diverse group described esoterically as a "design collective".

"My concept of design process is thus broad, broader than most would frame it, for those I take as members of a design collective are a varied lot. Participants may come from management, marketing as well as the structures group, the software department, or the electronics division.......even customers.....Any individual who has a legitimate say in the process, whose words, proposals, claims and supplications matter and contribute to the final form of the product I consider a participant" (Bucciarelli, 2002)

Increasing social dimension of Architecture-based analysis

Similarly the role of architects is continually being revised and expanded in light of their need to balance the individual interests of the ever expanding design collective. The consideration of non-functional properties includes the more traditional design considerations such as performance and availability but also opens the door on any number of imaginable attributes such as cost, time, usability, and safety, which naturally can all be reasoned about with relation to the structure of the system.

"When Brunel and Robert Stephenson were building railways in the late 1830s and 1840s, they were expected to involve themselves with raising capital, appearing before Parliamentary Committee, conciliating influential people...........Why should we be surprised if Software Engineers may need to draw on expertise in mathematics, financial analysis, business, production, quality control, sociology and law, as well as in each application area they deal with" (Jackson, 1995)

Jackson's software engineer as bricoleur is highly telling of the need to balance more than purely technical issues in engineering an effective system. Similarly when trying to evaluate what is, an effective system there needs to be adequate consideration of such concerns. ATAM and SARA, widely viewed as the industry best practice methods (Obbink et al., 2002) both strive to involve all key stakeholder groups, acknowledging the contribution of stakeholders to the realisation of an effective design and importantly achieving greater levels of "buy-in" from the group. In doing so these methods also bring upon themselves concerns associated with managing "the non-technical aspects of running an architecture review" (Kazman & Bass, 2002)".

The extent to which these concerns are understood and handled in the context of architecture-based analysis are conspicuous by their absence with only recent acknowledgement from Kazman, "as architecture reviewers, we continually run into social, psychological, and managerial issues and must be prepared to deal with them." (Kazman & Bass, 2002). He suggests resolution to these issues should occur through successful facilitation and process management, echoing several points from their literature about needing to negotiate your way into an organisation and effectively set expectations (Clements et al., 2002). Several pragmatic facilitation skills are also put forward as being integral for conducting a successful evaluation. Amongst these are the needs to "control the crowd, involve the key stakeholders, engage all participants, maintain authority, control the pace, and get concurrence and feedback".

While these behavioural aspects of group dynamics are important to the effective functioning of the group, they are insufficient in themselves to compensate for the effects of human factors within a process. Importantly they don't appear to explore the dimensions of complexity which arise when the social and psychological perspectives are taken into account

SHOULD ARCHITECTURE-BASED ANALYSIS BE PERCEIVED AS A 'SOFT' PROCESS?

Another perspective on social complexity

Design theoretic and methodological research offers another dimension to the characteristics of social processes, presenting the view that "we see reality through the mental filter of our 'ideas' or conceptions. If we accept this commonplace observation it is hard to see how one could ever talk about reality except through the very same filter." (Galle, 1999). Here Galle touches on a significant topic associated with human perspective and understanding, which has a well respected lineage in the form of 'Weltanshauungen' (Checkland & Holwell, 1998; Hitchins, 1992), 'holons' (Checkland & Holwell, 1998), 'psychological and metaphysical complexity' (Flood, 1988) and 'object worlds' (Bucciarelli, 1994).

In organisational development terms, the social system created by the collection of individuals needs to be considered as a soft system. Sir Geoffrey Vickers fostered the softening of hard systems thinking towards group dynamics. The previous view of organisations was that the group had a common goal and understanding and were working to achieve that goal through decisions. Soft systems thinking introduced the notions raised above about individual motivations, experience and views of the situation that needed to be both understood in context of their peer's world views and accommodated for in decisions (Checkland & Holwell, 1998). While it is reasonably logical to argue that architecture analysis does not possess an entirely congruent set of traits to that of an organisation it cannot escape the characteristics of being seen as a social process, akin to a "messy" human activity system (HAS) (Hitchins, 1992; Jagodzinski et al., 2000). The elements of hierarchy, different domains of concern, different historical perspectives and experience, different intentions (Galle, 1999), different perceptions of the situation (Janes, 1988), social disharmony, etc are all prevalent to the architecture analysis process, as much as they are the design process at large.

Compounding Factors

The nature of the artefact is not consistent with the nature of the task

When dealing with technology the temptation is to treat the process in the same light as the product. In Boulding's classification of systems, structures are "classified as physical or mechanical systems, i.e. hard, and are in the province of the physical sciences" (Hitchins, 1992). However the journey from concept (design need) to design artefact (communicative medium) (Bucciarelli, 2002) to system or structure does not resemble the characteristics of the end product at all. In terms of design, all that exists are representations of concepts of the system, which are in turn interpreted by the stakeholders (Galle, 1999). The use of design representations as a means of communication places the process at the 'social' end of Boulding's classification.

Specifying purely facilitator behavioural traits as the mechanism for managing social complexity within a process is noticeably dismissive of any need to adapt the process itself. The objectivity (Hilliard et al., 1996) and replicability (Kazman et al., 1994) that were the ideals of earlier analysis methods appear not to have changed. The same theoretical perspective that informed earlier beliefs about architecture-based analysis is still thought to hold even in the face of "psychological complexity" and the theoretical arguments about the nature of design (Galle, 1999).

Reasoning for such a perspective lies in the fact architecture has emerged amidst traditional 'hard' systems thinking processes (Jackson, 1988) where requirements of function are discovered and refined to exact system designs that perform the required functions. Function is a reasonably tangible way of measuring the conformance of a concrete system to requirements, either the function is performed or it's not. Architecture on the other hand deals with the abstract form of the system and similarly attempts to reconcile quality requirements in addition to functional requirements, which in many instances are themselves hard to produce metrics for and hence reason about in the context of a system structure.

In disciplines where the process is well bounded by normative rules and understanding such that measures, functions to manipulate those measures and refutable ways of utilising the outcomes, are all explicitly defined, there is perhaps a diffused impact of social complexity. Although there have been some attempts at relating structural measures to quality attributes (van Gurp, 2000), accompanied by the declaration of several design heuristics such as Attribute-Based Architecture Styles (ABAS)s (Klein & Kazman, 1999), it can be said that few irrefutable or un-situated truths currently exist in the architecture-based analysis world.

The nature of requirements

Figure 1 and Figure 2 depict architecture-oriented design life-cycles, in which architecture-based analysis is shown as being informed by a comprehensive requirements engineering exercise, however it is a fairly well respected belief within the software engineering community that requirements engineering exercises are fraught with uncertainty.

"...it is really impossible for a client, even working with a software engineer, to specify completely, precisely, and correctly the exact requirements of a modern software product before trying some versions of the product" (Brooks, 1987)

Adding to the requirements problem is the fact that quality attributes are a more recent concern in systems design and are commonly represented and reasoned about in a vague manner.

"In a perfect world, the quality requirements for a system would be completely and unambiguously specified in a requirements document .... In reality, requirements documents are not written, or are written poorly, or do not properly address quality attributes." (Kazman et al., 2000)

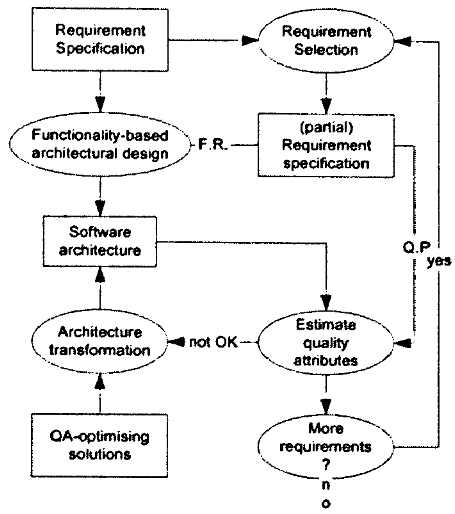

Figure 1 – Quality attribute oriented software architecture design process-(Bosch, 2003)

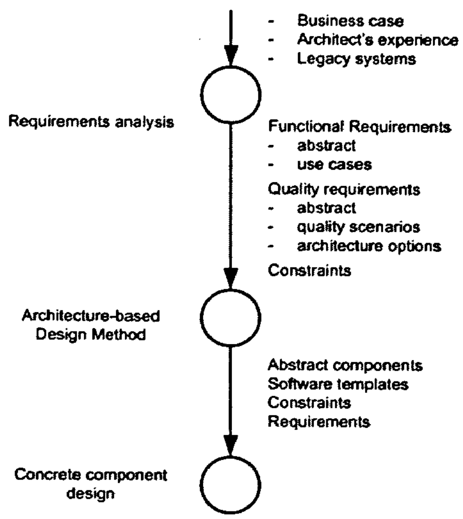

Figure 2 – ABDM (Bachmann et al., 2000)

DOES SOCIAL COMPLEXITY NEED ADDRESSING IN ARCHITECTURE-BASED ANALYSIS?

As is the case with Schön's architects of the built environment and their sketches, systems architecture deals in the realm of virtual worlds.

"The situations of Quist and the Supervisor are, in important ways, not the real thing. Quist is not moving dirt on the site. The Supervisor is not talking to the patient. Each is operating in a virtual world, a constructed representation of the real world of practice" (Schön, 1991)

Similarly in architecture-based analysis, the architecture presented to the stakeholders is a partial representation of the system, from which they are left with the task of mentally constructing the system, its goals and importantly their intent for it. These world views both unite stakeholders in some aspects and divide them in others, for the view that they share the same object worlds has already been rejected (Bucciarelli, 2002). Davies suggests that the metaphysical complexity introduced in situations such as examining complex virtual systems is dealt with in human terms by "human sense-making" simplifying the world by selecting from it "that which it takes to be important aspects of that world" (Davies, 1988).

"This is the selection of relevance from the world via an assimilation and accommodation process" [Piaget 1952 in (Davies, 1988)].

Soft systems methodology maintains that this accommodation needs to be reached in a group sense, through a common understanding of the system at hand and an appreciative understanding of the individual world views of the stakeholder group (Weltanschauungs) (Checkland & Holwell, 1998). Only when accommodations are made and a sort of group understanding formed, can the ideal system be reasoned about. Without this common understanding the individual contributions can conceptually swamp the process, imposing their view upon the situation and adding to the situational complexity rather than seeking to resolve it (Davies, 1988). Analogies can be drawn in the world of waves where harmonic waves interfere constructively and disharmony causes them to destructively interfere.

Practical evidence of the need to build analysis from a common understanding of the system is found within Bass' text on software architecture principles, where accounts of a design review and an architecture-based analysis review showed constructive argumentation. Initial questioning by one observer sparked the input from another, who offered further insight into an account of the repercussions of a design decision (Bass et al., 1998). Significantly for the evaluation process is discovery of the design problems from unstructured questioning of the architect in both cases. In one case the scenario was the springboard for the questioning however it was the interrogation-style perusal of the matter by a stubborn stakeholder that actually uncovered the problem.

Bass suggests that the stakeholders have a "limited" role of "helping craft the statement of goals for the architecture and then helping articulate scenarios" (Bass et al., 1998), perhaps by way of mitigation for any problems experienced by involving the stakeholder community. However the way in which the stakeholders view the system, their intended uses for it and their overall goals for the system are the critical benchmarks that drive the analysis methods. Understanding these factors with respect to the stakeholder group is imperative to the success of the analysis process, something which appears to be jeopardised by the existing lack of consideration for managing social complexity within the architecture-based analysis process.

Being the medium through which the stakeholders communicate, architectural representation is a logical nexus of viewpoints and concerns for the design process. Architecture evaluation acts as a key integrating component serving to both explore the problem space further by expounding the undeclared goals of the customer as well as provide guidance for the architects in attempting to realise a satisfactory solution.

As we recall the earlier discussion of Jackson's software engineer as bricoleur and the social behaviour guidelines for the ATAM it becomes evident that systems architecture has placed the responsibility for managing social complexity on the rather crowded shoulders of the architect, or in the case of architecture analysis, the facilitator/s. In many ways augmenting the importance of facilitation can be highly counter-productive to the process of building understanding. The SEI have noted the apparent "mismatch" that occurs in the communication chain of architecture-based analysis.

"even though the review team is frequently the focus of the conversation and the source of many of the probing questions. The review's outputs are really for the stakeholders—the review team members are just there to act as catalysts, experts, and facilitators. Because of this mismatch between the producers and consumers of the information and the way that the information is elicited (through the facilitation of the review team), extra care must be paid to ensure that all stakeholders concur..." (Kazman et al., 2000)

The review is essentially charged with juxtaposing the problem owner's position with that of the solution strategist, to ensure that they align. Architecture is the means through which they communicate and negotiate understanding of each other's object worlds, a negotiation that Bucciarelli argues needs to take place within a social framework (Bucciarelli 84 in (Sargent, 1994)). The concentration on representation in "architectural" terms and the focus of the communication on the facilitator has sought to conform a social situation with a highly rigid process, instead of a social framework.

HOW CAN SOCIAL COMPLEXITY BE MANAGED WITHIN ARCHITECTURE-BASED ANALYSIS?

In looking to control the social complexity associated with architecture-based analysis, research should focus on two main principals, born of the need to firstly construct the participants view of the system in a way that integrates all of the stakeholder viewpoints and secondly the need to balance the conflicting aspects of these viewpoints.

Within the context of 'social organisational' thinking a key concept to reasoning about the social complexity of group processes is the use of what can be termed methods of 'shared reality construction' (Truex et al., 1999). Which explores the notion that even constructed beliefs can be termed the existent reality, in the event that is agreed upon by the group. A concept reasonably sympathetic to the view that the system is tested against the norms of what the group wants it to be, not against some loftier notion of what a "good" system is. Therefore the essential task becomes converging the group viewpoint towards a common understanding of the goals, requirements and system they are meant to evaluating. Methods like interpretive structural modelling (ISM) help to represent complex, linked ideas in a form that is both palatable and understandable to the participants (Janes, 1988; Kanungo & Bhatnagar, 2002). Goal-based requirements (van Lamsweerde, 2001) can be considered a specialisation of ISM, where the semantic linkage between the conceptual nodes is one of "is achieved by", in a refinement context. As well as providing operational context to requirements, goals have the added advantage of offering a dimension of rigour to scenario elicitation. Other techniques such as building a common language and semantic are also highly important to the process of converging group understanding towards an integrated group perspective.

Modelling complex situations is a common goal in many disciplines, however very few of them handle the notion of plurality in an explicit manner. Soft Systems Methodology is one such process that has within its methodology a distinct aim of creating a common view of the system being examined through rich pictures, as well as aims to understand and reconcile diverse viewpoints through root definitions and balance these viewpoints within a single representation, conceptual modelling. Integral to coping with the social complexity, and pivotal to SSM is the need to accommodate disparate and often opposing stakeholder views. Progress and meaningful action in SSM are generated through accommodations, which essentially represent outcomes which are considered fairly balanced with respect to the polarity of group opinion.

CONCLUSION

The encouraging progression of architecture-based analysis from an expert-centric, evaluation focus to a stakeholder-centric, analysis focus has improved its utility in complex problem situations. Consequently it has unknowingly introduced several new dimensions to the complexity of the task itself, which are likely to impact directly upon its efficacy, by way of affecting the way in which the stakeholders view the system, their intended uses for it and their overall goals for the system.

By modelling different information elements in an esoteric fashion, peculiar to the responsible stakeholder sub-group and by assuming a common understanding of the system, its purpose and the most important elements thereof, architecture-based analysis doesn't seek to resolve the social complexity inherent in a diverse stakeholder gathering. Further to this, the seeming "mismatch" in communications and strength of facilitation, which has been suggested as a way of handling non-technical issues, have the capacity to further conceptually isolate the assembled stakeholders from each other.

Our research has established a compelling case for the existence of what has been termed 'social complexity', in architecture-based analysis. It has also discussed some appropriate methods of handling this facet of complexity through incorporating plurality into information gathering and representation, as well as utilising methods to balance opposing and potentially irreconcilable views. Future research will focus on applying these methods in a complex systems project, driven by an appropriate methodology capable of providing deep understanding of a practical learning situation. In this instance, action research will be utilised because of its capability to grant insight into situations where the issues are born of constructed knowledge in an essentially social context. The need to achieve meaningful change to the process in order to progress the project, the need for the researchers to act on the project itself in an instrumental capacity and the inherent need for iteration in complex systems projects encourages the application of Action Research, of which iteration, participation and reflection form important constituent phases.

REFERENCES

Abowd, G., Bass, L., Clements, P., Kazman, R., Northrop, L., & Zaremski, A. (1997). *Recommended Best Industrial Practice for Software Architecture Evaluation* (CMU/SEI-96-TR-025). Pittsburgh, Pennsylvania: Sofware Engineering Institute (SEI), Carnegie-Mellon University (CMU).

Bachmann, F., Bass, L., Chastek, G., Donohoe, P., & Peruzzi, F. (2000). *The Architecture Based Design Method* (CMU/SEI-2000-TR-001). Pittsburgh: Software Engineering Institute.

Bass, L., Clements, P., & Kazman, R. (1998). *Software architecture in practice* (1st ed.). Boston: Addison-Wesley.

Bengtsson, P., Lassing, N., Bosch, J., & Vliet, H. v. (2000). *Analyzing software architectures for modifiability* (HK-R-RES-00/11-SE). Högskolan: Karlskrona/Ronneby.

Boehm, B. W. (1981). *Software engineering economics*. Englewood Cliffs, N.J.: Prentice-Hall.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer, 21*(0018-9162), 61-72.

Bosch, J. (2003). Software Architecture - Engineering Quality Attributes. *Systems and Software, 66*, 183-186.

Brooks, F. P. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *Computer, IEEE, 20*(4), 10-19.

Bucciarelli, L. (1994). *Designing Engineers*. Mass: MIT Press.

Bucciarelli, L. L. (2002). Between thought and object in engineering design. *Design Studies, 23*, 219-231.

Budgen, D. (1994). *Software design*. Wokingham, England ; Reading, Mass.: Addison-Wesley Pub. Co.

Checkland, P., & Holwell, S. (1998). *Information, systems, and information systems : making sense of the field*. Chichester ; New York: Wiley.

Clements, P., Kazman, R., & Klein, M. (2002). *Evaluating software architectures : methods and case studies*. Boston: Addison-Wesley.

Davies, L. J. (1988). How SSM deals with complexity. *Transactions on Instrumentation Measurement and Control, 10*(3), 130-138.

Dobrica, L., & Niemela, E. (2002). A survey on software architecture analysis methods. *Software Engineering, IEEE Transactions on, 28*(7), 638-653.

Flood, R. L. (1988). Situational complexity, systems modelling and methodology. *Transactions on Instrumentation Measurement and Control, 10*(3), 122-129.

Galle, P. (1999). Design as an intentional action: a conceptual analysis. *Design Studies, 20*, 57-81.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns : elements of reusable object-oriented software*. Reading, Mass.: Addison-Wesley.

Hauser, J. R., & Clausing, D. (1988). The house of quality. *Harvard Business Review, 66*(3), 63-73.

Hilliard, R., Kurland, M., Litvintchouk, S., Rice, T., & Schwarm, S. (1996). *Architecture Quality Assessment, version 2.0*: MITRE.

Hitchins, D. K. (1992). *Putting systems to work*. Chichester ; New York: Wiley.

Houkes, W. (2002). Design and use as plans: an action-theoretical account. *Design Studies, 23*, 303-320.

IEEE. (1999). IEEE standard for application and management of the systems engineering process, *IEEE Std 1220-1998*.

IEEE. (2000). IEEE Recommended practice for architectural description of software-intensive systems, *IEEE Std 1471-2000* (pp. i-23).

ISO/IEC. (2002). Systems Engineering - System life cycle processes, *ISO/IEC 15288:2002*. http://www.jtc1-sc7.org/: International Organization for Standardization.

Jackson, M. C. (1988). Systems methodologies as complementary tools for managing situational complexity. *Transactions on Instrumentation Measurement and Control, 10*(3), 155-160.

Jackson, M. J. (1995). *Software requirements & specifications : a lexicon of practice, principles, and prejudices*. New York

Wokingham, England ; Reading, Mass.: ACM Press ;

Addison-Wesley Pub. Co.

Jagodzinski, P., Reid, F. J. M., Culverhouse, P., Parsons, R., & Phillips, I. (2000). A Study of electronics engineering design teams. *Design Studies, 21*, 375-402.

Janes, F. R. (1988). interpretive structural modelling: a methodology for structuring complex issues. *Transactions on Instrumentation Measurement and Control, 10*(3), 145-154.

Joseph, S. (1996). Design systems and paradigms. *Design Studies, 17*, 227-339.

Kanungo, S., & Bhatnagar, V. (2002). Beyond Generic Models for Information System Quality: The use of Interpretive Structural Modeling (ISM). *Systems Research and Behavioural Science, 19*, 531-549.

Kazman, R., Barbacci, M., Klein, M., Jeromy Carriere, S., & Woods, S. G. (1999). *Experience with performing architecture tradeoff analysis*. Paper presented at the Software Engineering, 1999. Proceedings of the 1999 International Conference on.

Kazman, R., & Bass, L. (2002). Making architecture reviews work in the real world. *Software, IEEE, 19*(1), 67-73.

Kazman, R., Bass, L., Abowd, G., & Webb, M. (1994). *SAAM: a method for analyzing the properties of software architectures*. Paper presented at the Software Engineering, 1994. Proceedings. ICSE-16., 16th International Conference on.

Kazman, R., Klein, M., & Clements, P. (2000). *ATAM: Method for Architecture Evaluation* (CMU/SEI-2000-TR-004). Pittsburgh: Software Engineering Institute.

Klein, M., & Kazman, R. (1999). *Attribute-Based Architecture Styles* (CMU/SEI-99-TRR-022). Pittsburgh: Software Engineering Institute.

Kruchten, P. B. (1995). The 4+1 View Model of architecture. *IEEE Software, 12*(0740-7459), 42-50.

Land, R. (2002). *A Brief Survey of Sotware Architecture* (Report). Vasteras: Department of Computer Engineering, Mälardalen University.

Land, R. (2002). *Improving quality attributes of a complex system through architectural analysis-a case study*. Paper presented at the Engineering of Computer-Based Systems, 2002. Proceedings. Ninth Annual IEEE International Conference and Workshop on the.

Lopez, M. (2003). Application of an evaluation framework for analyzing the architecture tradeoff analysis method. *Systems and Software, 68*(3), 233-241.

Obbink, H., Kruchten, P., Kozaczyynski, W., Postema, H., Ran, A., Dominick, L., Kazman, R., Tracz, W., & Kahane, E. (2002). *Software Architecture Review and Assessment (SARA) Report.*

Parnas, D. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM, 15*(12), 1053-1058.

Pressman, R. S. (1996). *Software Engineering: A Practitioners Approach* (2nd ed.): McGraw-Hill.

Sargent, P. (1994). Design science or nonscience. *Design Studies, 15*(4), 389-402.

Schön, D. A. (1991). *The reflective practitioner : how professionals think in action.* Aldershot, England: Arena.

Shaw, M., & Garlan, D. (1996). *Software architecture : pespectives on an emerging discipline.* Upper Saddle River, N.J.: Prentice Hall.

Stevens, W. P., Myers, G. J., & Constantine, L. L. (1974). Structured Design. *IBM systems Journal, 13*(2), 115-139.

Truex, D., Baskerville, R., & Klein, H. (1999). Growing systems in emergent organisations. *Communications of the ACM, 42*(8), 117-123.

van Gurp, J. B., J. (2000). *SAABNet: Managing qualitative knowledge in software architecture assessment VO -.* Paper presented at the Engineering of Computer Based Systems, 2000. (ECBS 2000) Proceedings. Seventh IEEE International Conference and Workshopon the.

van Lamsweerde, A. (2001). *Goal-oriented requirements engineering: a guided tour.* Paper presented at the Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on.

Zachman, J. A. (1984). A framework for information systems architecture. *IBM systems Journal, 26*(3).

ACKNOWLEDGEMENTS

The authors wish to thank the ARC and Alcatel Australia for their continued support.

COPYRIGHT

Expanding the view on Complexity within the Architecture Trade-off Analysis Method

David Colquitt[1], John Leaney[2]
1. Faculty of Engineering
2. Institute for Information and Communication Technologies
University of Technology, Sydney
[David.Colquitt | John.Leaney]@uts.edu.au

Abstract

The following paper presents the learning outcomes from an investigation into the aspects of complexity involved in architecture-based analysis. Using a framework of situational complexity as provocation, the manifestations of complexity observed in the Architecture Tradeoff Analysis Method (ATAM) process are presented in terms of a people and systems dimension. These aspects of complexity are shown to impact upon some of the most important ATAM objectives. The change in ATAM complexity is also presented with respect to the design lifecycle. Some resolution to the complexity suffered by the process is suggested in terms of splitting out the analysis objectives and maintaining two types of analysis, as well as paying attention to the content aspects of the process that drive its direction from within.

Keywords
Software architecture, architecture analysis, situational complexity, ATAM

1. Introduction

Empirical [1] and theoretical [2] work both support the notion that the further along the system design and development path, the more committed stakeholders are to the solution and the more costly it becomes to change design decisions [3]. The tangible saving in terms of both effort and money has created a decisive need to reason about the finished systems properties using a baseline of only the earliest design artefacts. Software architecture (SA) is one such discipline that seeks to make use of abstract design representations, encouraging communication amongst stakeholders and providing a vehicle for reasoning about the design from the earliest system form onwards [4]. By establishing a relationship between the goals of a system, be they functional or non-functional and the underlying structure of the system, SA suggests the capability to design for desired properties.

The need to design for specific properties also requires a method by which these properties can be tested for in architectural designs, giving rise to architecture-based analysis techniques. Architecture-based analysis developed recently during two distinct epochs, the first marked by questioning techniques oriented towards candidate selection [5], the second marked by methods more focused on analysis throughout the design process [6]. Coincident with the change in analysis role was the expansion of the participating stakeholder group, showing a shift from expert-centric to stakeholder-centric analysis. The resulting group of participants can be considered, as they are in the broader design process, a *human activity system*, granting architecture-based analysis many of the attributes of a *social* or '*soft*' process [7].

Expanding the stakeholder group heightens the situational complexity, which Flood suggest comprises a 'classic/systems' dimension and a 'people' dimension [8]. Until recently the focus has remained on the systems aspects of architecture-based analysis with only recent acknowledgement of the people dimension. "as architecture reviewers, we continually run into social, psychological, and managerial issues and must be prepared to deal with them." [9]. The different world views (Weltanschauung) [10] of each participant will naturally effect how they view the task and participate therein. Until now the responsibility for handling this aspect of complexity has been placed in the hands of the facilitator. This is notably dismissive of the need to alter the process itself and the potential for the facilitator themselves to contribute to the complexity of the situation. The following paper presents an experience-based account of researching the complexity of architecture-based analysis within a combined industry-university design project.

Section 2 presents the importance of the research methodology and the research situation to the interpretation of the outcomes, as well as introducing the chosen architecture-based analysis method. The aspects of complexity identified within the Architecture Trade-off Analysis Method (ATAM) are then presented (Section 3) and their impact on the ATAM process discussed (Section 4). Learning from applying the process at different stages in the design lifecycle is then discussed in Section 5 and a brief conclusion given in Section 6.

2. Research Background and Methodology

2.1 Research situation

The research was undertaken as part of a broader project whose aim was to develop a proof of concept design for a next generation network (NGN) management system. Traditionally telecommunications infrastructure has been strongly engineered for quality, meaning that a relatively static topology and offline configuration of the network by-in-large accounts for the quality perceptions of the user. However faced with a significant decline in the value of its traditional services, the telecommunications market has perceived the need to enable new value added user services. The desire to provide innovative new service sets to consumers has created a step-wise increase in the complexity of management systems and placed them in the critical path for service delivery. In many ways the business capability is now seen as being defined by the capability of the management systems [11]. This significantly augmented the quality expectations of a class of systems that were already considered in the domain of complex systems.

Consequently the quality focus of software architecture and the telecommunications problem seemed a natural fit, realising a linkage project between the university and an industrial partner who was a leading vendor of telecommunications equipment and services. The project group consisted of academics from within the university, senior technology consultants from the industry partner and several doctoral researchers. The academics were drawn from different telecommunications and software disciplines. Their roles included oversight of each of the key project discipline of Architecture, Policy, Networking and Service. Each discipline had at least one associated doctoral researcher.

The research focus of this paper is a subset of the work undertaken as part of the Architecture practice of the team. While the issues addressed in this research are pertinent to the broader practice of architecture-based design as they are the specific area of architecture-based analysis, the focus was chosen due to reasons of involvement and clarity. The researchers were heavily involved in the Architecture practice of the design team. The issue of clarity refers to the ability to clearly identify the research area. Architecture-based analysis is a contained event that occurs within architecture-based design practice with a clearly defined method. Whereas the design team were consciously not following a structured design approach due to the nature of the research challenge.

2.2 The Architecture Trade-off Analysis Method (ATAM)

Architecture-based analysis techniques fall into one of two categories, questioning and measuring according to whether they offer qualitative or quantitative results. In complex design situations the effort required to develop models suitable for quantitative analysis and the concentration on one quality at the expense of others tend to dissuade the use of measuring techniques. While many of the subsequent questioning techniques provide the ability to evaluate multiple quality aspects of a system and don't require quantitative models, they still tend to only find application as candidate selection methods once a design has reached maturity [5].

The adoption of an iterative incremental development process required a method which could be used throughout the systems lifecycle, as well as provide insight into the design issues and how they relate to the customer objectives. Consequently the methods suited to such an approach are those oriented towards application from an early point in the design life-cycle as well as providing the ability to analyse the relationship between multiple quality concerns and design decisions. The only methods found to satisfy these conditions included Software Architecture Assessment using Bayesian Networks (SAABNet) [12] and the Architecture Tradeoff Analysis Method (ATAM) [13]. Although it is viewed as qualitative in nature, SaabNet requires the numeric coding of relationships between design aspects as conditional probabilities and as such requires determinism in the relationship between design moves and system properties that is not known.

The Architecture-based Trade-off Analysis Method (ATAM) was selected as the most appropriate for the research project as it could be used throughout the design lifecycle, achieved design analysis rather than candidate selection and had a strong lineage of development backed by case reporting. The method itself is broken into two overlapping phases of 9 steps as shown in Figure 1.

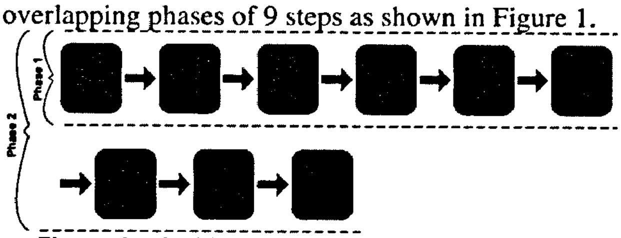

Figure 1 - Architecture-based Tradeoff Analysis Method (ATAM)

The first step introduces the method itself to the participants. This is followed by two steps presenting both the business case and the solution architecture respectively. The 4th step looks to identify key architectural approaches responsible for system qualities. The 5th step creates the attribute utility tree, which refines the business drivers into quality goals into concrete scenarios representative of the goals. The final steps identify architectural sensitivity (architectural decision key to a specific quality) points, trade-offs (architectural decision key multiple qualities) and risks (important decisions not made) [13]. (need to potentially add a diagram to this).

2.3 Action Research and Interpreting the Research Outcomes

As discussed in Colquitt [6], the original research interest in the "social" dimension of complexity promoted the idea there are difficulties the process has to overcome which are only attributable to the people within a situation. That is the problems arise from the intersection between the technical and non-technical [14], the interaction of the perspective of the stakeholders with the task of architecture-based analysis. The human dimension of which urges the use of qualitative methods to capture the complexity of the phenomena [14].

The need to act as a researcher in the telecommunications research project and simultaneously research the architecture-based analysis of systems promoted the use of a methodology that would accommodate both roles. Action Research (AR) as a methodology for situated inquiry is sympathetic of the need to perform both roles and is also accepting of change as a mechanism of developing further understanding. The need for both action and learning is revealed in the structure of the methodology which in its most abstract form consists of stages of planning action, taking action and reflecting upon action. The phases form a natural cycle, in which the reflection and learning from the previous cycle influences the planning and action to be taken in the next phase, as theory and practice both inform one another [15]. The particular method adopted for applying Action Research is that of Susman [16] Figure 2.

While Action Research reports on experience it needs to be understood as a more structured approach to inquiry than recollections of past experience. The research was undertaken with a specific aim to understand the complexity of ATAM; a specific method of inquiry (Action Research) and a defined framework of ideas representing the outcomes of background research. However the structure and discipline does not grant generality of the findings, and although the issues can be extrapolated out to large systems design the outcomes would be specifically different. The research should be understood in the spirit of Schon's "science before the fact" [2]. With the outcomes providing the basis for more controlled experimentation in the future.

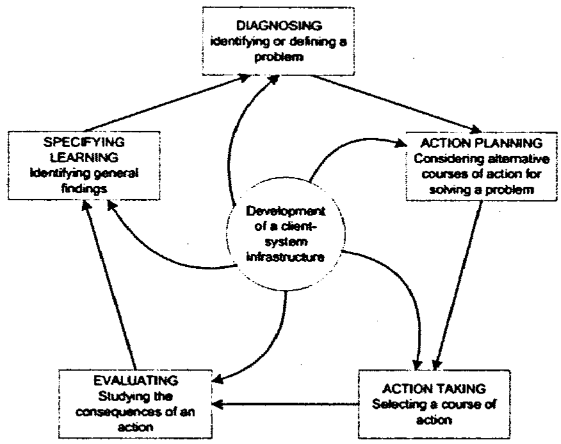

Figure 2 – Susman Action Research Model [16]

Importantly the content of the rest of this paper only seeks to present the outcomes of the research, not the processes whereby AR assisted in formulating the outcomes. Similarly for reasons of brevity some supporting material has been omitted in terms of design artefacts and quotes.

3. Complexity in the ATAM

3.1 An Expanded View of Complexity within the ATAM

The motivation for a specific focus on architecture-based analysis arose from an early project need to perform analysis on existing system designs. The research team exhibited strong diversity from the outset, and matters of design often became side-tracked by clarifying interpretations and viewpoints amongst the design group. Discussions indicated the role of the stakeholder in the project, their experience and areas of interest all contributed to the ways in which they participated. The importance of personal viewpoints, beliefs and interests aligns closely with what has been termed a 'people' dimension to complexity [8]. Importantly this aspect of complexity will increase as the perceptions and beliefs of the participants diverge.

The chosen method of ATAM promoted stakeholder participation, yet didn't openly tackle how to address the impact of stakeholder diversity. Instead the ATAM literature suggests these issues can be resolved by properly setting the expectations of the participating parties, ensuring documentation is made available and that the facilitator is sufficiently skilled [9]. The focus on the facilitation is dismissive of the need to adapt the process itself to handle such diversity. It also re-enforces the view that traditional systems engineering is focused on the 'systems' aspects of complexity at the expense of the 'people' dimension [17]. Complexity within a situation arises from both the number of parts and relationships of the system at hand, as well as the people within the design situation. Consequently the following sections present the aspects of both 'people' and 'systems' complexity encountered within the ATAM. Section 4 will discuss how this complexity impacted what the method was trying to achieve.

3.2 People dimension to complexity

People complexity suggests each individual's capabilities, beliefs and interests will influence how they participate in design situations. These aspects of complexity formulate the perspective from which a participant views the design situation. This perspective influences the way design artefacts are interpreted and communicated amongst the group. The term Weltanschauungen (W) [10], meaning world view is commonly used in Information Systems (IS) literature to express the idea that artefacts and actions can be interpreted in different ways according to each individual's perspective. Significant evidence for differing world views was found throughout the ATAM exercises.

3.2.1 Differing world views. The second and third steps of the ATAM are designed to present the business case for the system and the system architecture. These presentations are given by the industry partner (client) and the design team respectively. The difference in language and concepts used within these presentations provided the first insight into the perspectives at work within the analysis. The business drivers presented by the client emphasised the needs of cost management, including both operational and capital measures, as well as customer choice and satisfaction. These drivers are all contingent on the way in which the system is designed but are nonetheless, quite distant from the quality attributes such as performance and security commonly put forward when talking about system architectures.

The difference in language and concepts exhibited in the presentation activities of the ATAM also carried through into the quality attribute workshops. These workshops are designed to elicit the key quality attributes of the system that form the first tier of the utility tree. The quality responses of the industry partner used telecommunications business concepts such as "network optimised" and "customer aware". Alternatively the broader group tended to re-enforce the systems quality perspective adopted by the software architecture community, offering qualities like "performance" and "availability". However rather than simply re-enforcing the dichotomy of business and systems quality, the quality responses also indicated a more personal value-based view of the quality needs for the system. Vocal advocates of standards offered qualities like "standards-based", similarly those with broader experience in billing and mobile aspects to networks raised notions like "roaming" and "billing accuracy".

The language used to define the attributes themselves also reflected the personal notions of the participants. Those with backgrounds in telecommunications management used terms such as "protocols", "managed data", "5 9's" and "Frame Loss Rate". Participants with more hands on networking experience referred to "rack space", "moves/adds/changes (MACS)". Whereas the architecture-minded amongst the group used well known architectural constructs such as "connections" and "components" [18], the extreme of which was a systems architect who clearly defined 3 qualities then offered the view that these three were defined/contributed to by the remaining qualities.

3.2.2 Influence of Roles and Beliefs on Participation. The impact of personal beliefs on how individuals exercised their roles within the analysis also became quite evident through the project. In developing group artefacts there is an inherent aim to accommodate the views of all participants present. This accommodation creates a natural tension between group consensus and personal opinion. Participation in these social situations is based upon appreciative acts. Appreciative acts concern both judgements of reality (what is the case?) and judgements of value (is this acceptable or unacceptable?) [19]. These judgements cannot be ego-less and are intrinsically linked to the participant's view of their role in the situation. There were several instances of roles and judgement shaping individual participation.

The belief that qualities can be hierarchically organised influenced the way in which the lead architect, then facilitator went about utilising the quality responses. Instead of tallying the votes, the facilitator decided to use the quality relationships to group responses in a bid to include a broader range of quality attributes rather than simply selecting the most popular. Quality attributes are a key concern of software architecture and the groupings attracted the attention of a member of the architecture team, who sought to change the end result of the exercise.

Similarly in building the utility tree, the lead designer raised issue with assumptions about the system creeping into scenarios. Being the designer, any aspects of design that crept into the development of artefacts would directly impact their work and potentially force decisions that they themselves were unwilling to make at that stage. On each of these occasions, the impact of group processes on personal roles prompted remedial action by a participant.

3.2.3 Negotiating of Meaning as Critical. Given the diversity of language and concepts discussed in section 3.2.1 it is not surprising that the negotiation of meaning was an important theme throughout the research. Open dialogue helped surface assumptions and grow each participant's understanding. However this was seldom a controllable, brief exercise. Discussions of systems quality commonly branched into other areas of design and into business strategic considerations. Meaning was rarely just about semantics but a roundtrip through an associated network of concepts.

The hard won nature of negotiated meaning means that it is strongly defended. On separate occasions different participants sought to augment the set of quality concerns with a quality they saw as important following the completion of the quality attribute workshops. While the group could see the importance of these attributes they had not passed through the same process of group negotiation and couldn't be viewed as representative of the group's viewpoint. In the instance where changes were made to artefacts outside of the group processes, there was a significant degree of scepticism towards the changes. Some more damaging changes resulted in disinterest within the broader group.

3.3 'Systems dimension' to complexity

Whilst the initial focus of the research was the effect of the people dimension on complexity, there was commonly found to be contributing factors from the systems dimension. Be it due to the intractable nature of design, the conceptual nature of the system at such an early stage of design or the various ways a system can be decomposed,, it became apparent that behind most people problems, complex systems aspects could be seen to be co-incident.

3.3.1 Concomitant nature of the problem and solution. One of the most prominent aspects of systems complexity in the ATAM proved to be the relationship between the business-strategic and system requirements perspectives. Exploration of either perspective seemed to require knowledge from the other in order to understand it. For example considerations of the impact of specific quality requirements on a system commonly reverberated back to considerations at the business strategic level. Similarly in attempting to resolve answers to questions at a business strategic level, knowledge of the capability of the system was often sought. The problem and solution appear to evolve together and become concomitant. In software design the learning loop is perceived as taking place between the requirements and the design artefacts [20]. While this is indeed necessary and true, experience here has shown that the requirements embody an approach that attempts to resolve a business need for the client. Therefore the loop of learning between the original motivating problem and the approach lies as much between the aspirations of the client and the driving requirements of the system as it does between requirements and design activities. Potentially it is even more critical at this stage since the loop of learning bridges world views as well as from problem to solution (as section 3.2.1 attests).

3.3.2 Divergent nature of understanding. The difficulty in reconciling these viewpoints lies in developing a complete understanding of them. The search for solutions cannot be exhaustive due to sheer number of permutations in complex systems [21]. Experience from a goal-based requirements (GBR) workshop to bridge the business strategic perspective and systems quality perspectives helped highlight this aspect of complexity. Several goal-graphs of up to eighty nodes were produced, which only represented the higher level considerations.

3.3.3 Difficulty developing usage aspects. Another of the consistent difficulties in attempting to communicate aspects of the solution or problem was the elusive nature of use, or how the system would be used. Early on in the design lifecycle the system architecture is incomplete, hindering attempts at understanding the potential usage aspects of the system [22]. Even in the event that a complete functional structure was to be available there is still some doubt as to whether this adequately reveals the context of use [23].

Additionally no real precedent for such a telecommunications system existed. Therefore notions of use which would develop from detailed system knowledge were unclear. In their place abstractions of use, in this case the operational aspects, or operational specifications, were put forward. The problem with operational specifications like those so commonly used in telecommunications is that they are solution agnostic. They specify what has to be done but don't give clarity on how it should be achieved. A task like the ATAM really requires the structural detail behind how things are achieved to understand the quality ramifications. Two systems could quite readily exhibit the same operational characteristics but have two entirely different systems (structurally, architecturally) implementing them.

Further obstructing the understanding of usage was an expectation of innovation. The project has been conceived with the intention that the NGN management framework would supersede existing management practises. As such traditions and experience became largely invalid because they were perceived as coming encumbered with the past mistakes.

3.3.4 Environmental Turmoil. Complicating matters was the speed with which important environmental influences could change. Telecommunications is an rapidly evolving industry where technology and carrier behaviour is constantly changing. From within the organisation there were multiple company acquisitions and new patents brought to the design situation. The social, political and technological forces influencing the project made it difficult to stabilise the linkage partner's position, exacerbating the difficulty understanding the strategic and system quality associations.

4. Impact of situational complexity for the ATAM process

4.1 Disconnect between the business strategic and systems quality perspectives

Where aspects of systems and people complexity discussed in section 3 are coincident upon a process it is understandable they would affect the conduct and outcomes. Perhaps the most enduring of these affects was the difficulty to associate and understand the business strategic and systems quality perspectives. The relationship and understanding between these perspectives is fundamental to the aims of what the ATAM is designed to achieve. [4].

The ATAM literature offers a fairly close relationship between these two informational elements, in many instances proposing what are more commonly recognised as quality attributes, as business drivers. "For example, in an e-commerce system two of the business drivers might be stated as: "security is central....and modifiability is central to the success of the system..."" [13]. Contrary to these examples, the earlier discussion of the 'people dimension' to complexity established these as two different perspectives (world views). Importantly the actors aligned with these perspectives are likely to use significantly different language and concepts to express the driving system need. The diversity of these viewpoints caused difficulty for the ATAM in two main ways. The first was in understanding and utilising the responses. The second was building the business goal to system quality relationships necessary for constructing the utility tree.

4.2 Common understanding of quality viewpoint

The ATAM literature states the potential mismatch in communications resulting from the business owner and designers having to communicate through an intermediary in the form of a facilitator [9]. When confronted by the relatively unique quality terms of the linkage (industry) partner the facilitator sought to interpret them in more popular systems quality terms. The perspective was an important one, yet was unlikely to receive much representation in the process while it differed so significantly from the broader group. The experience affirmed the legitimacy of the systems quality perspective and the alignment of the facilitator's world view with it. Instances like this where participant's views are challenged in the face of broader quality frameworks have the potential to ostracise participants from a process where their input is critical.

Conducting the exercise from a specific world view can also stifle more diverse viewpoints. Or more worryingly suppress a unique insight which has not yet been appreciated by the rest of the group. Early on in design situations the problem is ill-structured and the more creative solutions challenge the brief rather than fulfil it [21]. The ATAM may be the first occasion stakeholders have to express specific insights they have to each other. Participants commonly showed significant interest in each other's qualities when they were tabled.

The constructive dialogue shows the importance of negotiating meaning [24], which should precede any negotiation of objectives or goals. Negotiating meaning is something that is only lightly raised in the ATAM. Although it professes the need for facilitators to seek concurrence and feedback [9], there is little suggestion as to what this is. Meaning in terms of the ATAM is primarily "negotiated" through the construction of the utility tree, where ambiguity of meaning is resolved through concrete scenarios. As discussed earlier in this section, by this stage the intended contribution of the participant may have been significantly altered to conform to quality attribute norms. Similarly scenarios proved no refuge for understanding as early on the usage context of the system is not well understood and interdependencies with business strategic issues can hamper the development of more detailed design concepts. This represents nothing of the social framework in which the negotiation of meaning should take place [24].

By not openly negotiating meaning throughout the process, ATAM tends to leave the participants isolated in their own perspective, rather than permitting them the ability to evolve their understanding of the situation with respect to each others views. This can significantly enhance the analysis process in both the emergent outcomes from constructive understanding as well as the less tangible aspects of providing the group with a common identity and understanding. The 'taken as given' meaning of individual perspectives [10] in contrast to group consensus was perhaps no better exemplified by a glossary exercise undertaken between analyses, in order to define problematic terms. It took several workshops and significant effort before the group was able to agree upon a definition for the word "service" alone. Notably the time taken to establish such meaning is not looked upon kindly where project schedules are important.

4.3 Disjoint between systems quality and business drivers

4.3.1 Assumptions in bridging perspectives exemplified by Quality of Experience. While the systems quality and business strategic perspectives proved to be quite different, the utility tree requires causal attributes be made between them in order to focus the analysis. Characterisations could be attempted in order to draw relationships between the business and quality viewpoints. However assumptions generally have to be made in order to do so. For example in the NGN solution space, systems commonly refer to the need for high levels of throughput and performance to ensure customer satisfaction [25]. Yet there are no guarantees that a performing system will be the determining factor in the customer's view of the service. The customer may be happier with a low performing cheaper service, or a service delivery method that does not have any real-time implications. The concept of Quality of Experience (QoE) acknowledges that the customer does not just use technology but lives with it [26]. Consequently quality aspects associated with the usage perspective partially influence, but aren't solely responsible for the customer perception [27]. Care needs to be taken in testing the assumptions behind framing the problem in a particular way [2], which is effectively what these characterisations are inviting participants to do. Viewing the customer satisfaction as largely a network performance issue narrows out of view other contributing factors to customer experience like ubiquity, cost, cultural appeal, etc [28].

4.3.2 Exhaustive Understanding of Requirements is Infeasible. Attempts at understanding the relationship between the systems quality perspective and business strategic perspective are complicated by the concomitant nature of understanding between them. Section 3.3.1 outlined how understanding of the business and technical solutions were dependant upon one another. Seeking to resolve the problem by reaching an exhaustive understanding of either perspective is likely to be infeasible given the combinatorically explosive nature of search through the solution space [21], exemplifying the divergent nature of real world situations [2]. Modelling the customer goals highlighted this divergence showing that quality attributes were implicit in some of the goals but their subsequent refinement provided no guarantees of yielding explicit quality attributes akin to those commonly found in software architecture literature. Furthermore the depth of reasoning (some 6-8 layers of hierarchy) highlighted how much refinement logic was internalised by the goal to quality associations.

4.3.3 Analysing against systems quality does not necessarily ensure customer satisfaction. The extent and complexity of the strategic viewpoint and the problematic nature of characterising goals as system qualities calls into question whether the system as judged from the quality viewpoint, satisfies the business strategic goals. From the experience of this research, this is largely not the case. Although the ATAM addresses important quality concerns there is no certainty these concerns ensure the satisfaction of the customer's business objectives or allays their greatest worries. The lack of certainty affect's the confidence of the group when deriving the attribute utility tree, which is effectively the centrepiece of the analysis. When faced with a utility tree devoid of business context, the participants sought to elaborate the utility tree with aspects of the business drivers. On reconciling the business and systems quality aspects, apprehensions were expressed as to the rigour of the relationship. The associations were re-analysed and changed according to the new consensus. The ready acceptance of change highlighted how unconvincing the original relationship was in that the group was comfortably able to reason through many changes.

5. ATAM and the design lifecycle

5.1 Design stages and their affect on the ATAM process

The previous sections presented the elements of situational complexity found to affect the ATAM and discussed the consequences for the process. The exercises that contributed to this learning all occurred across a significant time frame of the project. This granted insight into the use of ATAM both early on and throughout the design lifecycle. The following reflects learning on the relationship between ATAM and the design lifecycle.

Early in the design lifecycle quality requirements generally represented broader issues within the business drivers. The dependency of the system qualities on the business objectives saw a strong focus on clarifying the customer goals. The rationale for these goals appeared to be most strongly influenced by the external environment of the business. The constant pace of technological change and the ability for systems to define market capability continuously challenge the business to redefine itself in line with its environment[29]. These environmental factors, as well as scarring past experience are largely what the customer brings to the design situation [21]. Additionally the personal perspectives of the stakeholders ensure there is any number of views of the situation early on. At this stage in design the stakeholders are largely conceptually isolated and the focus of early activity is on negotiating meaning.

Once meaning has been negotiated the group can meaningfully discuss the quality aspects of the system in a bid to move from the business strategic to more systems design considerations. Personal meaning starts to become associated with group beliefs as negotiated artefacts are developed in group situations. Although in this design situation showed these beliefs were still confused by the lack of clarity surrounding the shape of the system design and its usage context. Importantly this experience only braces the early conceptual stage of design and successful ATAMs recounted within existing case reporting indicate that once the design is of sufficient maturity, usage concepts become embodied along with detailed behavioural understanding of the system [30]. The detail of the design helps reveal usage aspects and focuses the group on technical challenges, testing against what are considered to be fixed notions of strategic direction.

Figure 3 attempts to depict the progression of the design and the associated characteristics of analysis situation. Boehm's spiral model [31] is used as its overlay to indicate the early stages of design close to the origin and the later stages of design towards to the outer layers.

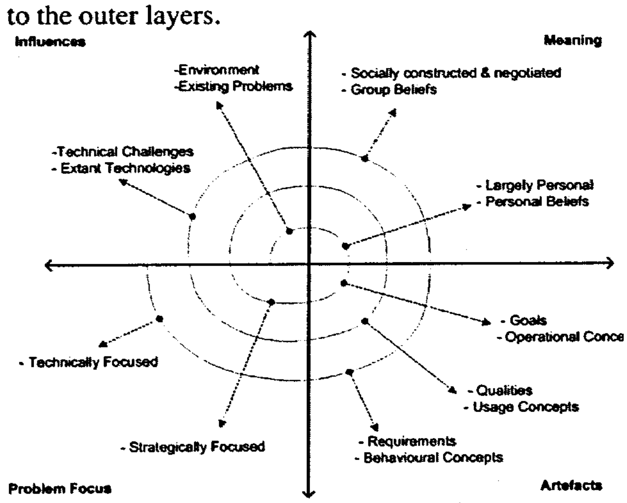

Figure 3 - Stages of design and its influence on group process

Evidenced by the experiences throughout this project as depicted in Figure 3 , the use of ATAM early on is problematic for several reasons. The first is that the ATAM does not focus on making sense of the business strategic viewpoint. Rather it is more focused on using it to develop a list of business drivers. However there are business problems to be solved as much as there are technical ones. Secondly it seeks to move quickly from the business strategic viewpoint into the systems quality one, which is actually a significant evolution in design terms. Lastly emanating from the quick progression from the business perspective to the quality one, there is an assumption of group understanding. The difficulty is that this has neither been given the open forums in which it needs to develop, nor the structure of detailed artefacts upon which it is commonly built.

The ATAM relies on well reasoned and stable understanding of the business strategic needs. Problematically these needs are exposed to the constant turmoil of the external environment and tend to develop in concert with, rather than well ahead of the system design.

Ideally the concomitant nature of the problem and the solution supports the need to deal with both at the same time. An approach that could potentially permit analysis at an early stage is to tailor the process according to the stage of the design. Figure 3 conveys the idea there are different needs at different times in the design lifecycle. Initially the problem focus is strategic; the strongest influences are previous system failures and changes in the business environment; meaning within the group is largely experience/belief-based and personal; and the system is represented by abstract artefacts such as goals and broad operational concepts.

Dealing with modelling the system goals exhibited very similar traits to system qualities, in that they were generally abstract, needed refinement to be properly understood and exhibited interdependencies, akin to design trade-offs.. These similarities imply the potential for an early ATAM exercise focused on the use of goal-level artefacts to establish the consequences between different strategic approaches. This would yield significant knowledge on business strategic issues permitting the customer stakeholder sufficient, rather than partial representation in the process. It also helps model the customer problems that tend to consistently upset the technical design process. The knowledge in this area could then be used to feed into the more technically focused ATAMs once the system has developed significant maturity, which sees the design stakeholders as receiving sufficient representation. Separating these two types of analysis allows the customer to analyse business issues with the same rigour that systems analysis covers design issues. The outcomes of the business focused analysis would help provide a firm basis upon which to build the quality requirements for the systems analysis. The business analysis also provides a rigorous mechanism to confront the constant change in the external environment, rather than continuously exposing the system design to the business uncertainty.

5.2 The importance of 'content' versus 'process'

In addition to the need to consider two different but inter-related types of analysis during the design lifecycle, there also needs to be a greater consideration for what analysis activities are achieving. In looking at design methodology Dorst noted that although most methodologies were specified in terms of processes, it was largely the "content" of the design situation that dictated the designer's actions. "In most cases, considerations linked to the content of the design situation (the perceived design problem, the designer's goals and the perceived possibilities for the next step) will determine the 'kind of action' (process-component)" [32].

It is therefore not surprising that the reality of systematic design conflicts with the way in which it is prescribed [21]. Similarly the ATAM concentrates heavily on the process itself at the expense of the content issues, which can significantly affect an analysis. For example is the goal of the presentation of the business context just that? Or is it to reach agreement and understanding within the group of the strategic issues influencing the system development and to carefully derive a set of system characteristics that can be seen to satisfy the strategic needs. Not forgetting of course there are multiple perspectives here. Do the designers have few expectations of the business context presentation and participate simply as a passive audience? Or do they expect a comprehensive presentation of issues availed to them through their interaction with the client, which they can rigorously question and seek to understand in a detailed way?

Where the reality of the activities does not fulfil the expectations of a stakeholder, there is little guidance for how resolution may occur. In this project improvisation took the form of glossaries, elicitation of meaning where it was not required, goal-quality matrixes and extensive elicitation of goal artefacts, amongst other things. Although explicit methods were used here to explore the improvisation there is a distinct danger that in commercial settings, with greater time pressures, where the facilitator would feel pressured to maintain in "control" of the exercise, the improvisation may well take the form of internal judgement and assumption. The danger of which is a self-sealing process [33], whereby the internal assumptions of the participants are not tested and any incidents that lurch towards difficulties in understanding are avoided lest they hamstring the entire process. However it must be acknowledged that this research can only point to the potential for this to occur due to the experience in the linkage project.

6. Conclusion

The ATAM represents a significant evolution for architecture-based analysis techniques. In place of masking the analysis process in the problematic scoring and manipulation of figures with a perceived end of candidate selection, the ATAM has taken responsibility for assisting understanding throughout design. The inclusive nature of the process also ensures that communication amongst the stakeholder community is enhanced. However as discussed in this paper the inclusive nature of the process and the conceptual nature of architectures challenges the evolved methods, particularly early on in the conceptual stages of design. The resultant situational complexity impedes some of the key objectives that the ATAM seeks to achieve, such as improved communication and a relationship between systems quality and stakeholder goals.

The impacts on the process can be seen to arise from the structure and perspective of the ATAM and notably extend well beyond the scope of facilitation. The process itself needs to adapt in order to provide the social framework in which the negotiation of meaning and objectives takes place. Currently the diversity of viewpoints are more likely to smooth over diversity than to openly encourage it into the process. However the uncertainty the richness of such diversity was found to impact on the design task which struggles to progress when exposed to constant change.

Consequently two streams of analysis have been proposed as a means of isolating the design perspective from the constant change of the business strategic environment. Further enhancement is also proposed through a greater focus on the content aspects of the system, which will drive participation from within the analysis aside from the external structure of the process imposed from outside it.

7. Acknowledgements

Funding for this work was furnished in part by the Australian Research Council (ARC) in the form of an Australian Postgraduate Award (APA) scholarship for David Colquitt. Additional funding and support has also been made available through an ARC Linkage Grant (grant number LP0219784), and the industrial partner, Alcatel Pty Ltd..

8. References

1. Brooks, F.P., *The mythical man-month : essays on software engineering*. Anniversary ed. ed. 1995: Addison-Wesley Pub. Co. xiii, 322 p.
2. Schön, D.A., *The reflective practitioner : how professionals think in action*. 1991, Aldershot, England: Arena. x, 374.
3. Abowd, G., et al., *Recommended Best Industrial Practice for Software Architecture Evaluation*. 1997, Sofware Engineering Institute (SEI), Carnegie-Mellon University (CMU): Pittsburgh, Pennsylvania.
4. Bass, L., P. Clements, and R. Kazman, *Software architecture in practice*. 2nd ed. SEI series in software engineering. 2003, Boston: Addison-Wesley. xxii, 528.
5. Lee, J. and K.-H. Hsu, *Modeling software architectures with goals in virtual university environment*. Information and Software Technology, 2002. **44**(6): p. 361-380.
6. Colquitt, D. *The case for understanding social complexity in the architecture-based analysis process*. in *Qualitative Research in IT & IT in Qualitative Research*. 2004. Griffiths University: Griffiths University.
7. Jagodzinski, P., et al., *A Study of electronics engineering design teams*. Design Studies, 2000. **21**: p. 375-402.
8. Flood, R.L., *Complexity: a Definition by Construction of a Conceptual Framework`*. Systems Research, 1987. 4(3): p. 177-185.
9. Kazman, R. and L. Bass, *Making architecture reviews work in the real world*. Software, IEEE, 2002. **19**(1): p. 67-73.
10. Checkland, P. and S. Holwell, *Information, systems, and information systems : making sense of the field*. 1998, Chichester ; New York: Wiley. xv, 262.
11. Hope, M. and E. Nichols, *OSS: New Approaches for Successful Service Management*. 2002, OVUM.
12. van Gurp, J.B., J. *SAABNet: Managing qualitative knowledge in software architecture assessment VO -*. in *Engineering of Computer Based Systems, 2000. (ECBS 2000) Proceedings. Seventh IEEE International Conference and Workshopon the*. 2000.
13. Kazman, R., M. Klein, and P. Clements, *ATAM: Method for Architecture Evaluation*. 2000, Software Engineering Institute: Pittsburgh.
14. Seaman, C.B. and V.R. Basili. *An Empirical Study of Communications in Code Inspections*. in *Proceedings of the 19th international conference on Software engineering*. 1997. Boston, Massachusetts, United States: ACM Press.

15. Avison, D., et al., *Action Research.* Communications of the ACM, 1999. **42**(1): p. 94-97.
16. Susman, G.I. and R.D. Evered, *An Assessment of the Scientific Merits of Action Research.* Administrative Science Quarterly, 1978. **23**: p. 582-603.
17. Jackson, M.C., *Systems methodologies as complementary tools for managing situational complexity.* Transactions on Instrumentation Measurement and Control, 1988. **10**(3): p. 155-160.
18. Perry, D.E. and A.L. Wolf, *Foundations for the Study of Software Architecture.* SIGSOFT Softw. Eng. Notes, 1992. **17**(4): p. 40-52.
19. Checkland, P., *Soft Systems Methodology: A thirty year retrospective.* Systems Research and Behavioural Science, 2000. **17**: p. S11-S58.
20. Fischer, G., D. Redmiles, and L. Williams, *Beyond Object Oriented Technology: Where Current Approaches Fall Short.* Human-Computer Interaction, 1995. **10**: p. 79-119.
21. Joseph, S., *Design systems and paradigms.* Design Studies, 1996. **17**: p. 227-339.
22. Brooks, F.P., *No Silver Bullet: Essence and Accidents of Software Engineering.* Computer, IEEE, 1987. **20**(4): p. 10-19.
23. Kroes, P., *Design methodology and the nature of technical artefacts.* Design Studies, 2002. **23**(3): p. 287-302.
24. Sargent, P., *Design science or nonscience.* Design Studies, 1994. **15**(4): p. 389-402.
25. Bouch, A., M.A. Sasse, and H. DeMeer. *Of Packets and People: A User-centered Approach to Quality of Service.* in *Eighth International Workshop on Quality of Service. IWQoS 2000.* 2001. Piscataway, NJ, USA: IEEE.
26. Keinonen, T. *User experience with communication technology.* in *Telework.* 2001.
27. Sheridan-Smith, N., J. Soliman, and D. Colquitt. *Improving the user experience through adaptive and dynamic service management.* in *Australian Telecommunications Networks and Applications Conference.* 2004. Bondi Beach, Sydney.
28. Armbruster, H., *Information infrastructures and multimedia communications: different approaches of the telephone, data, and radio/TV worlds.* Communications Magazine, IEEE, 1997. **35**(9): p. 92-101.
29. Truex, D., R. Baskerville, and H. Klein, *Growing systems in emergent organisations.* Communications of the ACM, 1999. **42**(8): p. 117-123.
30. Clements, P., R. Kazman, and M. Klein, *Evaluating software architectures : methods and case studies.* SEI series in software engineering. 2002, Boston: Addison-Wesley. xxv, 323.
31. Boehm, B.W., *A spiral model of software development and enhancement.* Computer, 1988. **21**(0018-9162): p. 61-72.
32. Dorst, K. and J. Dijkhuis, *Comparing paradigms for describing design activity.* Design Studies, 1995. **16**(2): p. 261-274.
33. Argyris, C., *On Organizational Learning.* Second ed. 1999, Massachusetts: Blackwell Publishers. xv 464.