Proceedings of QualIT2004: International Conference on Qualitative Research in IT & IT in Qualitative Research:
24-26 November 2004, Brisbane, Australia. Hosted by the Institute for Integrated and Intelligent Systems, Griffith University.
Copyright the authors and QualIT. Keywords: System architecture, Architecture-based Analysis, Evaluation Techniques, Soft Systems Theory, Social Issues.

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).

Method Name	First Published
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)

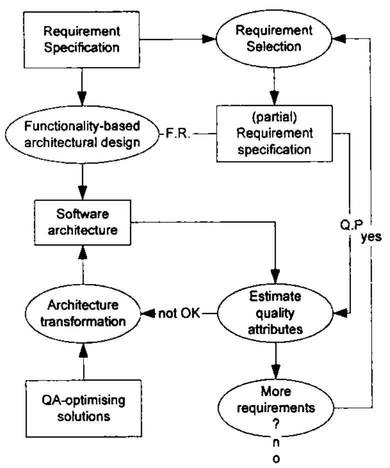


Figure 1 – Quality attribute oriented software architecture design process-(Bosch, 2003)

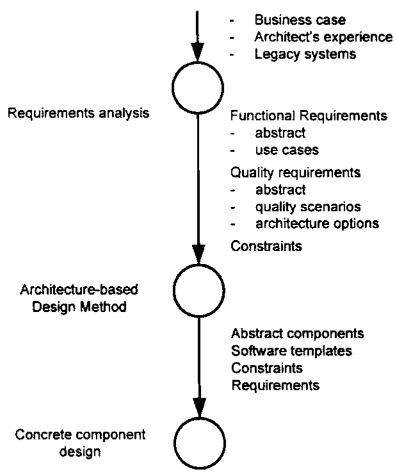


Figure 2 – ABDM (Bachmann et al., 2000)

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

2004 QualIT The way forward

International Conference on Qualitative Research in IT and IT in Qualitative Research

Griffith UNIVERSITY

Institute for Integrated and Intelligent Systems

QualIT 2004 Conference Proceedings

Software engineering

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

Authors: **David Colquitt**, John Leaney, Tim O'Neill

Full Text is available in PDF format.

2004

QualIT

The way forward

Date

November 24-26, 2004

Location

Brisbane, Australia

Qualitative Research in IT & IT in Qualitative Research

Institute for Integrated and Intelligent Systems (IIIS), Griffith University

Proceedings of the

Qualitative Research in IT & IT in Qualitative Research

Date:
November 24 - 26, 2004

Location:
Brisbane, Australia

Program Chair:
Jenine Beekhuyzen, School of Computing & Information Technology, Griffith University

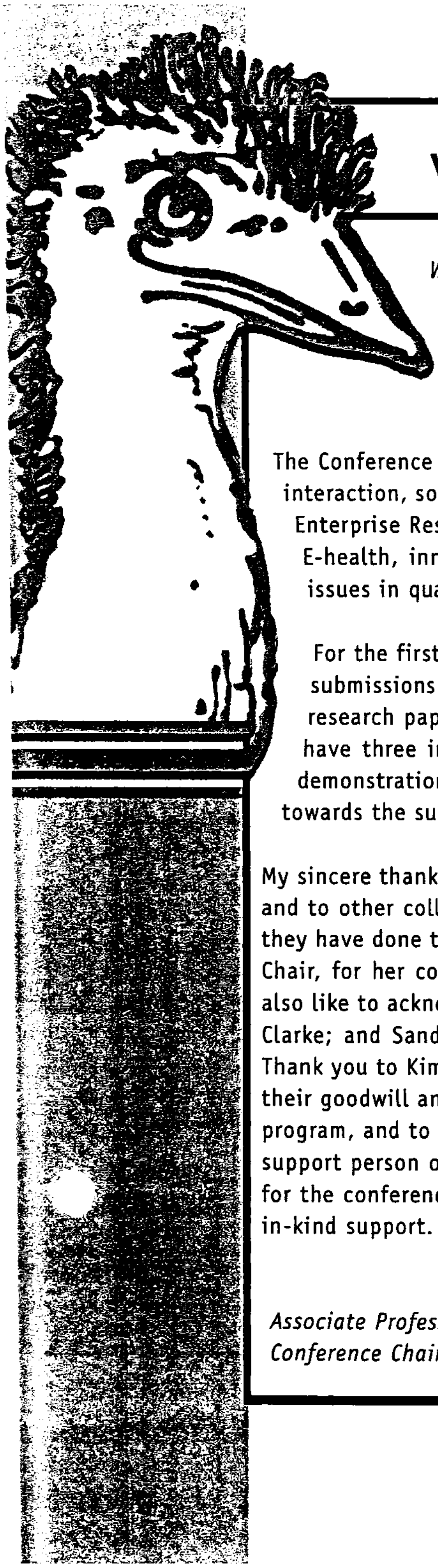
Conference Chair:
Associate Professor Liisa von Hellens, School of Computing & Information Technology, Griffith University

Conference Proceedings CD:
Karen Guest, School of Computing & Information Technology, Griffith University

Conference Web Site:
Michelle Morley, School of Computing & Information Technology, Griffith University

ISBN 1-920952-07-1

2004 QualIT
The way forward



Welcome from the Conference Chair

Welcome to QualIT 2004 and to Brisbane, Queensland, Australia.

This International Conference on Qualitative Research in IT and IT in Qualitative Research is the first time it has been held. I hope you enjoy the Conference, the location, and the Queensland summer climate.

The Conference brings together quality speakers of important and relevant topics including human computer interaction, socio-technical aspects of IT, IS implementation, software engineering, knowledge management, Enterprise Resource Planning, Emerging technologies, usability & User Centred Design, gender issues in IT, E-health, innovative qualitative studies, strategic use of qualitative software tools, and methodological issues in qualitative research to computing researchers around the world.

For the first ever QualIT Conference 65 submissions were received for the Conference program. Forty-two submissions (65%) were accepted for presentation at the Conference, including 38 research papers and 4 research papers in progress. All papers were double blind peer reviewed, unless otherwise mentioned. We have three international keynote speakers, four debating panels, two QSR workshops, and a Leximancer demonstration. We thank those involved in the panels and the workshops and appreciate your generosity towards the success of the Conference.

My sincere thanks go to my colleagues on the International Program Committee and the Organising Committee and to other colleagues not on the Committee for their dedication, assistance, and enormous amount of work they have done to put together the Conference. In particular, I wish to thank Jenine Beekhuyzen, the Program Chair, for her countless hours of organisation and co-ordination of every aspect of the Conference. I would also like to acknowledge student helpers: Renae Lewis, Kitty Kugyela, Jarrod Plant, Rebecca Dorries, and David Clarke; and Sandra Schneider who checked all papers making sure they were according to the right format. Thank you to Kim Taylor and Michelle Morley for their work on the website. I am most grateful to everyone for their goodwill and support. Thank you also to Toni Collier for the artwork for the CD and graphic design of the program, and to Anne Jackson for the wonderful Australiana designs; and to Mark Siedle as the main technical support person over the two days of the Conference. Thanks also to Lyn Richards for giving us the inspriation for the conference theme. Finally, I wish to thank the sponsors and exhibitors for their financial support and in-kind support.

Associate Professor Liisa von Hellens
Conference Chair