GSB WORKING PAPER NO. 30

FROM JAD TO INTEGRATIVE CONNECTEDNESS

BY

DES KLASS AND ALMA WHITELEY

© GRADUATE SCHOOL OF BUSINESS, CURTIN UNIVERSITY OF TECHNOLOGY, GPO BOX U1987, PERTH, WESTERN AUSTRALIA, 6001. TELEPHONE 61 8 9266 3366, FAX: 61 8 9266 3368
ISSN 1323-7853
ISBN 1 7406 7 288 7
JULY, 2003

CRICOS PROVIDER CODE NO. 00301J

Information in this publication is correct at the time of printing and valid for 2005, but may be subject to change. In particular, the University reserves the right to change the content and/or method of assessment, to change or alter tuition fees of any unit of study, to withdraw any unit of study or program which it offers, to impose limitations on enrolments in any unit or program, and/or to vary arrangements for any program.

No. 30
Working Paper Series
03.02

From JAD to Integrative Connectedness

by

Des Klass and Alma Whiteley
The Graduate School of Business,
Curtin University of Technology

© Graduate School of Business, Curtin University of Technology, GPO Box U1987, Perth, Western Australia, 6001. Telephone 61 8 9266 3366, Fax: 61 8 9266 3368
ISSN 1323-7853
ISBN 1 7406 7 288 7
July, 2003

CRICOS Provider Code No. 00301J

Abstract

Integrative Connectedness emerged from an IT decision theoretic software development research. Co-creation is a key to the developer/implementer relationship. "Buy-in" becomes more important than a "Buy the software” attitude. A Resource Allocation Modeling Process (RAMP) was the vehicle used to take the idea of joint application development (JAD) further, conceptually and practically. The theory of complex adaptive systems (cas), in particular the combination of if-then rules, anticipatory mechanisms and the space of the adjacent possible was used as a cornerstone of integrative connectedness.

Introduction

This paper focuses on the activities of software design and development, in particular as these relate to the user stakeholder group. As Liou and Chen (Liou & Chen 1993) point out some of the major problems in software design originate in some early misunderstanding or misinterpretation of systems requirements. As the authors say, there are many people typically affected by a software project in one way or another. This produces a “correction” mentality and apart from the cost (at least a hundred times greater than correcting them at the requirements definition stage), there is the lack of confidence and connectivity that would bind a client to a software developer.

It has traditionally been the case that there has been a relationship, often one to one between developers and users. As Liou and Chen observe, there has been a ‘one at a time’ type of interview preferred to collect data on requirements. What these authors suggest is that computer supported meetings (in this case using Group Support Systems (GSS) technology) are an improvement because of the opportunity for real time collaboration across a range of developers and users.

The idea of Joint Application Development (JAD) has brought the activity of software design closer to a range of involved parties. Liou and Chen describe some elements that demonstrate how JAD combats the errors they reports when using less inclusive systems design processes. JAD typically revolves around a comprehensive and facilitated workshop. There is a conscious aim to gain commitment from managers and other users. There is a sharing of design requirements and specifications and on the choice of structured methods and procedures to be used. These include process aspects like how to resolve future conflicts, how to keep communication going. They include substantive aspects and hard data collection is a workshop outcome. Alongside Joint Application Design (JAD) has developed the idea of Participatory Design (PD) (Carmel, Whitaker & George 1993). As indicated in the title, the notion here is on facilitated interactions between users and designers. Both JAD and PD have the propensity to elicit tacit knowledge and to encourage the sharing of knowledge and experience. Sometimes, as Carmel et al say, reporting on PD there is considerable trouble taken to ‘visualise’ the target workplace for which the software is being designed. Here, the use of figurative and metaphorical linguistic devices are used and these have a soft qualitative quality.

Whilst there are some important differences between JAD and PD, such as the definition of user, the inclusion/exclusion of managers, the styles of involvement and the actual techniques used (JAD is a structured process, PD is learning by doing) they both move systems design firmly into the involvement and participative arenas.

Three strong themes that come from both are comprehensiveness, inclusivity (user involvement) and a 'task supported by process' approach. "JAD represents a movement toward more collaborative practices to enhance the viability of given goals. In contrast, PD represents a movement toward more technical practices to enhance the viability of given social goals". Though both PD and JAD stress continuous involvement of the users and support user involvement of all parties affected by the information technology/systems, in practice, however, each approach excludes certain sets of affected parties.

This paper considers an enriching complement to both JAD and PD concepts (whilst recognising that they do have differences). Still taking a systems approach, some ideas from complex adaptive systems theory are presented.

Hisotrical Underpinnings

In this section, we chart the history of the software era, setting it against the backdrop of the modernist tradition as it circumscribed some of the paradigmatic assumptions within which the 'computer age' was set. We acknowledge with gratitude the contributions of Tapscott and Caston (1993) and Barley and Kunda (1992). Tapscott and Caston took us through software development eras and paradigms. Barley and Kunda described the ideological stances of rational and normative eras and paradigms.

First, we ask the question, why is software so important to business? As we will see, later in this section, the 'computer industry' was thought of synonymously with computer hardware. We argue here that some of the most inimitable and therefore most valuable assets an organization can have are intelligence, brainpower and creativity. Software is the expression of these qualities. We would argue also that in many cases, software can be a most valuable learning tool. The core concept of adaptive learning is trial and error (Argyris 1999). Argyris's (1999:22) criticism of Van de Ven and Polley's (1992) accounts of adaptive learning through technological innovation directs the thinking to the popular vogue of 'selling' software to stakeholders. The medical technology product was a process called the Therapeutic Aphaeresis Program (TAP).

> *[Before each review TAP managers spent a day] rehearsing their presentations, developing tactics and scripts on how they would respond to possible quotations of top managers…preparing slick*

> *visuals…reconstructing negative information in a positive frame, with assurances that they were in control of problems and presented action plans for addressing them*

As we can see here, there was not a true allegiance to the sort of reporting, negative as well as positive that would encourage an adaptive learning relationship to flourish. A second most important criticism for this paper was that of using statistical measures and hard data to replace 'reports from the firing line'. In other words a connectedness with stakeholders was conspicuously missing. So when we are asking about the contribution of software to business we must take into account the spirit of the software developer. If the software is developed in the spirit of 'sales and after service', in other words there is a software product or process and it is finished and sold and there the relationship with users ends apart from technological hitches, then the contribution to business may be valuable but short term. If, in contrast, there is a spirit of joining together, of connecting, then continuous value can be delivered as the learning will be more than the sum of the software parts.

Barley and Kunda (1992) report on the systems rationalist era that is around the mid-1950's to 1980's. "During World War II, the British and American military employed teams of mathematicians, physicists and statisticians to devise methods for solving logistical problems. ..Working with early computers, these 'operations research teams' were so successful that each of the services established its own operations research unit". Operations research (OR) as a concept, like its predecessor scientific management more than fifty years before, became instantly popular. The concept was built on scientific foundations, employing a realist ontology and a mathematical epistemic culture (Knorr-Cetina 1999). As Burack and Batlivala (1972 in Barley and Kunda (1992)) report, the two peak bodies in OR, The Operations Research Society of America and the Institute for Management Science, dedicated themselves to developing and applying quantitative techniques to management problems. These led to the accelerated development of computer science. Queuing theory, network analysis , simulation techniques, theories of linear and dynamic programming were developed without, it seems very much concern for the less quantitative, more elusive 'voices' of those who used the various computer systems.

This is not surprising, given the ambience of the managerial times. Even throughout the 1970's and 1980's when the human relations movement was well disseminated and widely acknowledged (Mayo 1945; McGregor 1960; Trist 1981; McGregor 1969), the command and control ethic flourished. "Process theories thereby presented management with a definition of itself consistent with the tools of OR and science. In fact, the process theorists were decidedly calculative even when they were not quantitative" (Barley and Kunda 1992:10). Such managerial ideologies must have had a significant effect on attitudes towards both decision makers and users connected with hardware and later software development. Systems rationalists was the name given to technology and computer systems designers. Below, we have developed a line of logic, simulating the way that thinking about systems design might follow for the systems rationalist.

Computer and systems design is expert work. It depends upon expertise from a number of scientific disciplines including mathematics, physics computer science and statistics. Users of technology and this would include hardware and software, were not expert. They needed to be educated within the limits of their need to operationalize the particular systems. They were, in a sense, part of the 'experiment' in the sense that a piece of hardware or software could not be said to be 'tested' until feedback on usage was collected. This feedback would be factual and would be linked to the specifications of the hardware or software. The feedback instrument would assume an instrumental (rather than developmental) motivation on the part of the user. The rhetoric would be in keeping with 'systems thinking' which itself was rational and exclusive of those outside of the systems thinking discourse boundary.

In charting some of the paradigmatic changes in information technology (IT), Tapscott and Caston (1993) identify three critical shifts in IT applications. One is the shift from personal to work-group computing. For designers, the focus had to change from 'a user' to 'a user group' or team. With this shift communication across a diverse group of users came into sharp relief. Goals related to those team effectiveness paralleled those pertaining to the hardware or software specifications. This meant incorporating interpersonal skills and communication into the systems designer tool-bag.

A second critical shift was that from what Tapscott and Caston call 'system islands' to those of integrated systems. The 'silo effect' is well documented in organizational literature (Semler 2000)and this term is used both to contrast with innovative and cooperative management activities and to describe the sort of functional fixedness that are often a consequence of organizational designs based on rational functionality. Systems islands are a kind of IT silo, the silos being management systems, financial systems and human resource systems. It would be fair to say that the development of IT architectures outpaced some of the paradigmatic shifts suggested required for open and cooperative management systems. The concept of the networked organizations is not new. Pascale (1990), Dunphy and Stace (1992), Dunphy and Griffiths (1998), Clarke and Clegg (1998) all suggest that this sort of organizational design needs to be the way of the future. Speaking with the IT voice, Tapscott and Caston (1993) characterize the change as below. This critical shift in organizational paradigm has direct implications for the ways in which hardware and software are disseminated into organizational structures and relationship methodologies. Later in the paper, the integrated connectedness methodology reflects the paradigm shifts below.

	Closed Hierarchy	**Open Networked Organization**
Structure	Hierarchical	Networked
Scope	Internal/Closed	External/ Open
Resource Focus	Capital	Human, Information
State	Static/Stable	Dynamic, changing
Personnel Focus	Managers	Professionals
Key Drivers	Reward and punishment	Commitment
Direction	Management Commands	Self-management
Basis of Action	Control	Empowerment to act
Individual Motivation	Satisfy superiors	Achieve team goals
Learning	Specific Skills	Broader competencies
Basis for Compensation	Position in hierarchy	Accomplishment, competence level
Relationships	Competitive (my turf)	Cooperative (our challenge)
Employee Attitude	Detachment (it's a job)	Identification (its my company)
Dominant Requirements	Sound management	Leadership

Source Tapscott, D. and A. Caston (1993). ***Paradigm Shift: The New Promise Of Information technology*****. New York, McGraw Hill.**

The third critical shift mentioned by Tapscott and Caston is that of the shift from internal to inter-enterprise computing. Just as the silo effects were a feature of rational and functional organization design, the concept could be applied between organizations. Now, it is not unusual to see Airlines connected to hospitality and tourism organizations as well as the most diverse range of service organizations that a traveler might need surrounding the journey. In other words the role of software has expanded, as more sophisticated and user-sensitive processes are required.

One of the most important interfaces between information systems and organizational development is the role of the employee. Often this is the person who, either individually or in a team, will implement a software process or program. Returning to Barley and Kunda (1992) and continuing with such writers as Lewin and Regine (1999) and Stacey et al (2000), we see that the 'supplying software to' or 'designing software for' concepts would not fit the emancipated role of contemporary employees. They have a need to make sense of the information technology with which they are presented and they need to do this within their organizational contexts. In other words the 'to' and the 'for' need to be replaced by the 'with'. This 'with' idea has indeed been incorporated into the software design rhetoric although how far this has moved away from the 'feedback' activity remains a question not too well answered. The following is an example of the incorporation of the "with" idea.

In the development of the software "Allocate" used in the study described later in this paper, the decision makers were required to provide criteria "weights" in a specific format that would allow the software to use a mathematically efficient method to determine the order of "Buy-in" for projects. The decision makers on the other hand considered "weights" implicitly and less formally and found the requirements of the software difficult to provide. With the insights gained by the implementers (decision makers) as to the software requirements on the one hand and the software developers appreciation of how the implementers dealt with their decision choices on the other hand, a new way of configuring the weights was jointly co-created and the software was reconfigured in a unique way that allowed implementers to provide the information in a form that did not compromise the integrity of the information required while at the same time enabling them to provide the information in a form they were comfortable with.

A well-known vehicle in software development is JAD, that is the joint application development activity. Developers of decision theoretic tools/software have traditionally focused largely on technical sophistication. Developers usually give some consideration and seek some knowledge about prospective users but at the end of the day, products are usually provided to the implementers (users) who are expected to reap the benefits that the technology affords (i.e. more sophisticated software).

Conceptually, what this conjures up is the notion of ‘application’. In other words, at the conception stage of the software development there is a tacit intention to “apply”. There is also sometimes a directional flow to the idea of ‘application’. There is often a powerful mathematical edifice underlying the software and the idea of communicating this is, correctly, not feasible or even desirable. However, this keeps the developers out of the ‘warmer’ and more interactive, even commonsense, arena of the customer context. It is plausible to suggest that this one-way product development stance of developers results in developing ‘ideal’ decontextualized products.

A more involved process design is evident in the JAD, PD and Rapid Application Design (RAD). With these processes, users are more involved, their views are heard, their commitment is sought and this undoubtedly makes them feel valued clients. We argue that this still does not allow for the movement into the “space of the adjacent.”

We turn now to Whiteley and McCabe’s (2001) model of ‘sustainable third wave change’. There are some elements in this model that resonate with the integrative and connectedness aspirations that were expressed in the empirical study described below.

Here we see senior managers who enfold co-creation of meaning into their strategic processes. What this means is that they recognise the need for strategy that is adaptive in nature, complex and most importantly fosters deep generative patterns of adaptive behavior. There would be an unpredictable quality in organizational strategies as well as coherence. Order would have an emergent quality. Thinking about managers and in particular here we draw attention to the team leaders or section heads who would be called upon to oversee software implementation, there would be two keynotes. One would be to foster continuous learning. The other would be to recognize the existence of and value of tacit as well as explicit knowledge. For both managers and employees, tacit knowledge needs to be incorporated into formal and specific arrangements. For those who implement software processes almost on a daily basis, the requirement under third wave change would be of a software that allowed guided self mastery, users to be self as well as expert- referring, an atmosphere of co-creation and, in some cases, the opportunity to be a genuine part of the software process continuous development undertaking.

Two theories that allowed us to incorporate some of the organizational and information technology paradigm shifts into our thinking were those of soft systems methodology (SSM) and complex adaptive systems (CAS). We present this theory as a prelude to describing an empirical study (Klass 1999) that set out to test and emerge some of the concepts we were theorizing about. In particular, the objective was to 'go beyond' JAD workshops and other such processes that were aimed at including the software user as an active participant in the design process.

Soft Systems Methodology

> *Regarding as a whole, the soft systems methodology is a learning system which uses systems ideas to formulate basic mental acts of four kinds:* perceiving *(stages 1 and 2),* predicating *(Stages 3 and 4)* comparing *(stage 5) and* deciding *on action (stage 6). The output of the methodology is thus very different from the output of hard systems engineering: it is learning which leads to a decision to take certain actions, knowing that this will lead, not to 'the problem' being now 'solved' but to a changed situation and new learning (Checkland 1999:17)*
>
> .

The interesting thing about soft systems methodology, developed by Peter Checkland (1999) is the thinking and assumptions upon which it is founded. Checkland incorporates into his own essentially scientific thinking, notions of what postmodernists (Cooper & Burrell 1998) would call the 'undecidable'. Human activities are recognized by Checkland. The idea of continuous learning is harmonious with the concepts of integration and connectedness that emerged as 'theories' of the respondents in the study reported here.

There is a playful and what Salzer-Mörling (1998) calls 'folkloric' tone to the methodology. When Checkland talks about 'world-images' he moves away from the supremacy of facts and incorporates intuition and imagination into his methodology. An outcome of this for software designers is that user or other stakeholders are seen as subjects rather than objects. Like Blumer, the symbolic interactionist theorist, Checkland assumes that people attribute meaning as they engage in human activity. In fact, Checkland is resonant with writers like Zohar (1997) who uses the quantum metaphor to talk about the 'particle' (or rational) and wave (or imaginative) qualities all humans have. The consequence is 'both and' thinking and when applied to methodologies such as SSM, nothing is lost. Although Checkland does not explicitly say so, there is a sense that the many paradoxes in human thinking would not present a problem. In terms of our thinking about the software development processes, his appreciation of all stakeholders as ‘thinkers and learners’ is very encouraging.

One of the cornerstones of soft systems thinking is the rejection of ‘problem/solution’ thinking. Applies to software design and development, this would mean that the software designer could not (and should not) consider the job done when the product or process is released. An often under-recognized benefit that even the humblest and technology slow user can bring is the opportunity for learning, not least that which comes from a new and fresh perspective. A theoretical way to state this is that the designer and the user would be invited into each other’s life worlds of meaning. Here we see an exchange relationship, a sharing of personal wisdom and schemata (Gell-Mann 1994). Together, given the opportunity, the designer and the user can produce a more complex account of the experience of designing and using software. The notion of complexity lies in stark contrast to that of the more simple and linear assumptions of scientific management (Taylor 1911) and rational systems Barley and Kunda (1992). This is, perhaps overstating the case as many writers in the systems thinking area refer to the ‘art’ that goes alongside the science of engineering Tapscott and Caston 1993). One thing that we can say with confidence was not evident in the literature, although many of the characteristics were present in soft systems thinking, were the non-linear and adaptive relationships we see within complex adaptive systems thinking.

Complex Adaptive Systems

> *…a complex adaptive system acquires information about its environment and its own interaction with the environment, identifying regularities in that information, condensing the regularities into a kind of “schema” or model, and acting in the real world on the basis of that schema. In each case there are various competing schemata and the results of the action in the real world feed back to influence the competition among those schemata(Gell-Mann 1994:17)*

Three visions enter the mind when thinking about the user role in software design. There is the traditional role where the user is in the software developers mind as a consumer of a ‘product’. The product is the technological software. This of course reflects The second vision is one of user involvement. The user is in the mind of the developer as an important source of information upon which to ascertain requirements and preferences about the software product. The requirement/preference focuses on the value-adding or the leverage gained through the use of the final product. The third vision is one corresponding to the elements in the model in figure 1., the integrative connectedness model. This vision is overtly relational. This means acknowledging that the value-adding or the leverage dimensions are not ‘pre-determined’ but discovered in an interactional fashion. Figure 1 adopts a complexity framework bringing into play several concepts developed at the Santa Fe Institute (Kauffman 1995; Gleick 1997; Holland 1995). The first concept was presented by Gleick on his work on Chaos- the concept is “order in Chaos”. What this means is there will be an inherent order to a situation and this order can be found in potentially chaotic environments. Software design is usually reported on in a somewhat linear, orderly undertaking. The “Chaos” (including that of relationships) is not an acknowledged aspect of software design. However, by adopting the concept of uncertainty, spontaneity and unpredictability (essentially more chaotic than orderly) software design can take on a new guise. In figure 1a) the designer will draw upon two dimensions, one is the certain knowledge (technical, factual and process) the other is potential knowledge. The two rules for potential knowledge according to Kauffman (Kauffman 1995) must be spontaneity and unpredictability. In moving from the certain knowledge to the uncertain or potential knowledge the designer crosses what we call the edge of uncertainty.

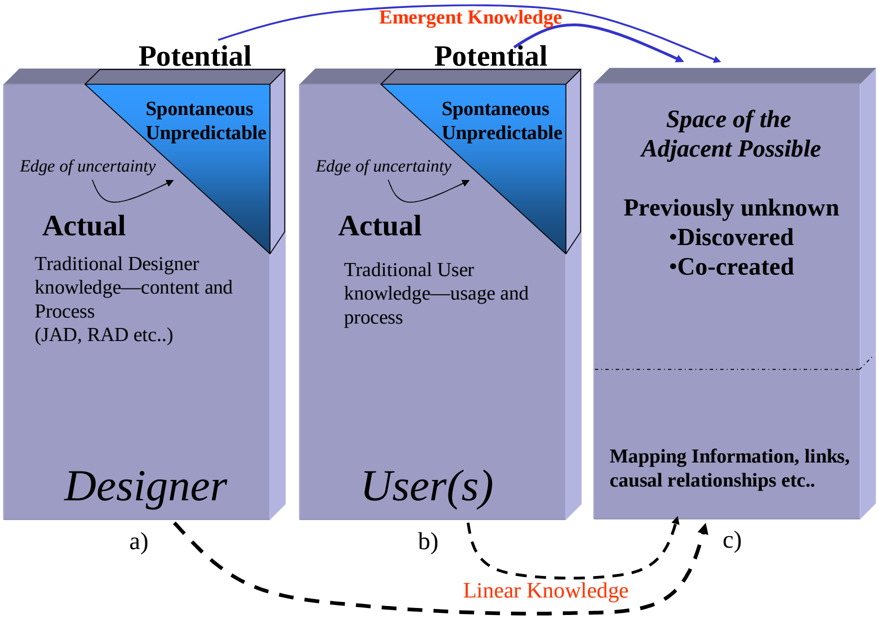

Figure 1. Integrative Connectedness --the Complexity framework

The same scenario applies to the user(s) –Figure 1b). The user(s) has knowledge on usage and process and also residing in the user(s) is spontaneous and unpredictable, yet to be discovered insights. As we move to figure c) we are only interested in the result of the potentials i.e. spontaneous, unpredictable and uncertain. Kauffman calls this thinking in the space of the adjacent possible

What we are suggesting is that we should be moving to part c in figure 1 by providing a process that would make possible the opportunity for co-creation. What happens in co-creation is the recognition that people develop internal rules or schemas. These rules represent the individual's capability to develop a working theory. Hitherto in the reporting on software design and development there is evidence that the designers and users have predetermined constructs that they bring with them into the JAD workshop and these constructs are the ones integrated into the software design. These predetermined constructs or schemata should be subject to constant reappraisal in the light of spontaneous conversation and the new insights gained from this. Holland would call these conversations 'working models' (we would call them 'the potential'). The danger lies when the participants in the designer-user partnership concentrate on extracting regularities from experience and gradually building them into sets of 'if-then' rules. The rules set up expectations and these are part of the anticipatory mechanisms utilised when responding, deciding or acting. Like the concept of the 'mental model'(Senge 1992), the schema provides a way of mapping a way through the complex information that has taken on bombardment proportions in recent years.

> *…we will view cas as systems composed of interacting agents described in terms of rules. These agents adapt by changing their rules as experience accumulates. In cas, a major part of the environment of any given adaptive agent consists of other adaptive agents, so that a portion of any agent's efforts at adaptation is spent adapting to other adaptive agents. This one feature is a major source of the complex temporal patterns that cas generate(Holland 1995:10).*

Whilst we acknowledge that mapping the way through complex information using mental models is an important and necessary part of the systems design activity there is an inherent danger of these mental models becoming self-fulfilling and self-reinforcing. We are proposing a both/and philosophy emphasising the notion of 'adaptive mental models'. In figure 1 we have introduced Kauffman's ideas of spontaneity, unpredictability and self-organization into the software design process. As individuals interact, (predicting as they go) they produce a potential. Not only do they do this but they do it interactively. The interactive process is co-creation. Kauffman (1995) uses the imagery of moving individual creations into "the space of the adjacent possible" (Griffin, Shaw & Stacey 1998) as they develop potentials together and co-create new pieces of meaning. We see here a generative approach to order and the journey into *cas* does not proceed very far without meeting several paradoxes. This acceptance of concepts such as generative order, spontaneity with prediction and embedded emergence is a key to the understanding of *cas.*

The Study

The authors have been involved for several years in the development of the computer software tool ‘Allocate’ (Klass 1997). After several applications of the tool it was discovered that a process was evolving around the software to allow for efficiency in data collection, better quality data, the enhancement of the likelihood of problem ownership and the improvement of the tool’s usefulness. Building on the process focus that was considered the ‘critical’ part of implementation, a management process was developed. In order to further investigate the importance of process a notion of combining the software ‘Allocate’ and an integrative management process, was conceptualized. Within this framework, a study was conducted to explore the acceptability and adoption of a resource allocation information technology process (Klass 1999). The study aimed to discover the constructs that may influence the acceptability and continued usage of the Resource Allocation Modeling Process (RAMP) (Klass 2000) intervention and to investigate the conditions that may influence decision makers’ perception of the usefulness of the RAMP intervention. The interest was in the performance of the management process. The study targeted decision makers within Western Australia local government who were responsible for resource allocation decisions.

The participants in the study shared common administrative backgrounds, had similar decision making activities and were new to the information technology (RAMP) used as the intervention in the study. A qualitative methodology was used within an interpretative epistemology and a constructivist ontology. The research design included 1) an introductory session where the ‘Allocate’ software was demonstrated together with an explanation of the process 2) the actual implementation of RAMP within the organization and 3) in-depth interviews with the implementers of the process. The value of the findings was more suggestive rather than definitive. Reflective thinking about what the implementers were saying led to the conceptualization of two different but connected expressions of developer / implementer relationship. One was very familiar in terms of traditional systems emphasis that was categorized in our model as a Technical Sophistication cycle. Here the journey for the implementer was charted from the need for a software solution to the software experiences. The impression gained from the findings led to a technological pathway that was linked to the client by outcomes. Another perspective and observation gathered in the in-depth interviews portrayed another picture. It was clear that key elements of a relationship would transform the 'buy' concept into a 'buy-in' concept. This 'buy-in' concept had inherent in it the dimensions of ownership, mutual commitment and the inclusion of these dimensions gave the respondent a holistic desire rather than a functional one.

The findings suggested that there is a difference between the valued client who will 'buy' and the valued client who 'buys in'. Figure 2 shows these two paths within the emergent model (Klass 1999). Path 1 represents the 'buy' cycle and Path 2 represents the 'buy in' cycle. In the 'buy in' model an attempt is made, through the various workshops and meetings of JAD-like techniques, to gather users' views and needs and exchange them with designers' software parameters and possibilities. Users' (and other stakeholders) needs are 'enfolded' into the product features. The data suggested that some of the elements of *cas* were intuitively sought by respondents and these will be discussed within the framework of the 'buy-in' concept. The notion of 'joint application' will be replaced by the concept of *integrated connectedness* (Klass 1999)

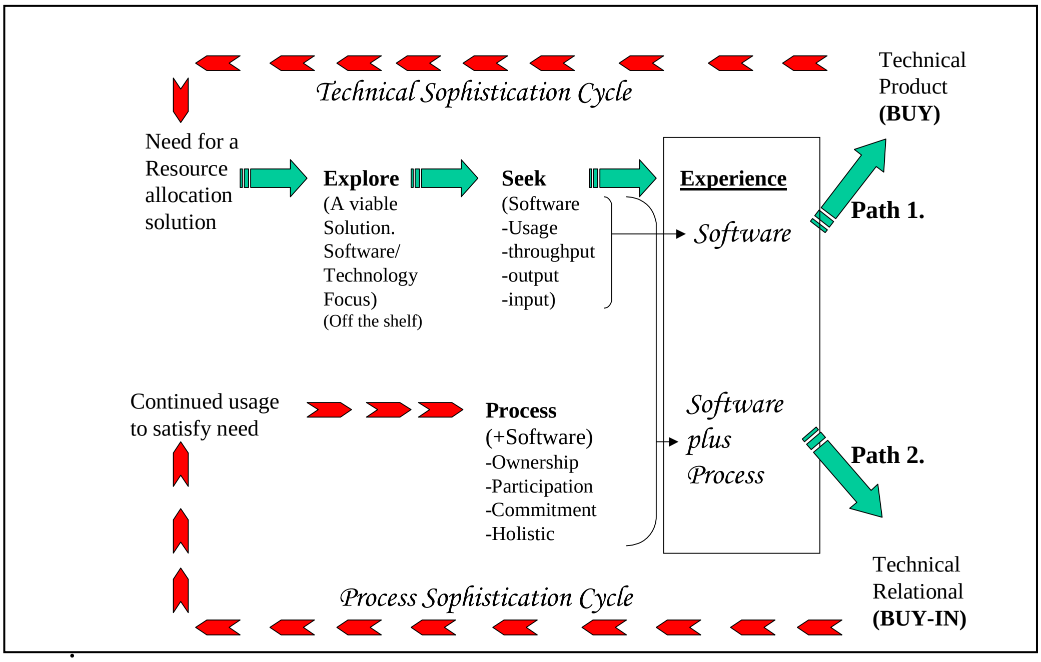
Technical Sophistication Cycle
Technical Product (BUY)
Need for a Resource allocation solution
Explore (A viable Solution. Software/ Technology Focus) (Off the shelf)
Seek (Software -Usage -throughput -output -input)
Experience
Software
Path 1.
Continued usage to satisfy need
Process (+Software) -Ownership -Participation -Commitment -Holistic
Software plus Process
Path 2.
Technical Relational (BUY-IN)
Process Sophistication Cycle

Figure 2.The process of adoption of a decision theoretic technological (RAMP) intervention. The Emergent Model (Klass 1999 p 234)

Discussion

The essence of the ‘buy in’ model presented here is that it is an *unfolding* or *becoming* model. Rather than capture a slice of the here and now or even extrapolate into the future there is an emergent and adaptive quality to the software management process. The formal data and anecdotal evidence from some of the ‘unfolding’ sessions held with the implementers in the study suggests that in traditional and participative processes there is much taken for granted in two distinct ways. Relating to such devices as JAD and PD, although activities are comprehensive and inclusive, there is little spontaneity and unpredictability reported in the process. Secondly there is little attempt to penetrate beneath the surface of each stake holding group in the sense of eliciting such things as anticipatory mechanisms and ‘if then’ rules.

That is not to say that some may not emerge as all well facilitated workshops encourage imaginal activities such as brainstorming and visioning. During these it is usual to find surprising and unpredictable data emerging. The leap made in this paper as a result of respondents’ contributions is that these can be better utilised to take a formal role in the sense making process. They can be recognised as ‘potentials’ and selected for co-creating together. This thinking leads to a reappraisal of brainstorming design and goals. If brainstorming is designed to generate a lot of ideas quickly and these are ‘rationalised’ through such devices as discuss /organize (a module within the software MeetingWorks^{TM} (Lewis 1993)) then it is unlikely that self-organization will be recognised. If the goals are task oriented and brainstorming or visioning fulfill that purpose, then it is unlikely that any inroads can be made into discovering the ‘if then’ rules (Holland 1995) and anticipatory mechanisms used by all participants in their sense making. Both Holland and Gell-Mann (Gell-Mann 1994) agree that these are fluid and changing on the basis of continuous appraisal and feedback.

There is a special sort of connectivity required. It needs to go beyond that of the participative process in the guise of one or more workshops and some continuing communication. The emerging ideas to come out of *cas* theory such as those discussed above, self organization, anticipatory mechanisms, if-then rules and adaptive agents would conjure up more an image of *unfolding* than enfolding.

In this study, the realisation came from listening to the data. Comments participants made struck a chord about ownership and of the importance of the management process embedded in the RAMP model (Klass 1999).

Integrative Connectedness

The term signifies a special sort of integration and connectedness between software developers and implementers. The proposed theoretical framework is at the formative stages of conceptual development although some preliminary research on 'if then' rules has already been conducted with some success (Whiteley & Wood 2000).

Integrated connectedness can be seen to be embryonic. It entailed first, a combination of if-then-rules (Holland 1995) and anticipatory mechanisms. The second is co-creation of meaning by moving the "potential unfolded by experience…into the space of the adjacent possible"(Griffin, Shaw & Stacey 1998).

The mechanisms for anticipation to which Holland (Holland 1995) and Kauffman (Kauffman 1995) refer provide another dimension to the individual's personal reality. If the individual decides on some action or outcome based on a final choice between his/her personal if-then rules then this will not necessarily gain commitment from others. It is this anticipatory quality that is a cornerstone of integrative connectedness.

We are suggesting that gaining commitment is a central goal of software implementation. It is argued that if it is possible to incorporate into individual sense making, in a practical way, the anticipated responses of others then commitment to the design will be enhanced. From the point of view of software development and implementation, the ideas taken from complexity theory and *cas* in particular have linked with the responses from the RAMP study. Both the spoken responses and the anecdotal confidences exchanged during the management process activities pointed to more than the straightforward needs for participation and collaboration. The sense of being part of each other's sense making and for looking for those potentials that suggested themselves but were not part of the current reality resonated strongly with the co-creation theory of Kauffman (1995).

References

Argyris, C. 1999, *On Organisational Learning,* 2nd edn, Blackwell Business, Oxford.

Barley, S. & Kunda, G. 1992, 'Design and Devotion: Surges of Rational and Normative Ideologies of Control in Managerial Discourse', *Administrative Science Quarterly,* vol. 37, no. September, pp. 363-399.

Carmel, E., Whitaker, R. & George, J. 1993, 'PD and Joint application design: A transatlantic comparison', *Association for computing Machinery. Communications of the ACM,* vol. 36, no. 6, pp. 40-49. Retrieved: June, from

Checkland, P. 1999, *Systems Thinking:Systems Practice,* Wiley, New York.

Clarke, T. & Clegg, S. 1998, *Changig Paradigms: The transformation of management though for the 21st Century,* Harper Collins Business, London.

Cooper, R. & Burrell, G. 1998, 'Modernism, Postmodernism and Organisational Analysis: An Introduction.' *Organisational Studies,* vol. 9, no. 1, pp. 91-112.

Dunphy, D. & Griffiths, A. 1998, *The Sustainable Corporation,* Allen and Unwin, Saint Leonards, Australia.

Dunphy, D. & Stace, D. 1992, *Under New Management: Australian Organizations in Transition,* McGraw-Hill, Sydney.

Gell-Mann, M. 1994, *The Quark and the Jaguar: Adventures in the simple and the complex,* Abacus, London.

Gleick, J. 1997, *Chaos: The amazing science of the unpredictable,* 2nd edn, Minerva, London.

Griffin, D., Shaw, P. & Stacey, R. 1998, 'Speaking of Complexity in Management Theory and Practice', *Journal of Organisation,* vol. 5, no. 3, pp. 315-339.

Holland, J. H. 1995, *Hidden Order: How Adaption Builds Complexity*, Helix Books, Addison-Wesley Publishing Co., Reading, Massachusetts.

Kauffman, S. 1995, *At Home in the Universe: The Search for the Laws of Complexity*, Viking, London.

Klass, D. 1997, *ALLOCATE Starter Guide*, Graduate School of Business, Curtin University of Technology, Perth, Western Australia.

Klass, D. 1999, *Decision Technology as a change intervention:A case of adoption using a resource allocation modeling process*, DBA, Murdoch University.

Klass, D. 2000, *RAMP: A Resource Allocation System*, Graduate School of Business, Curtin University of Technology, Perth, Western Australia.

Knorr-Cetina, K. 1999, *Epistemic Cultures: How the Sciences Make Knowledge,* Harvard University Press, Cambridge Mass.

Lewin, R. & Regine, B. 1999, *The Soul at work: Unleasing the Power of Complexity for Business Success*, Orion Business Books, London.

Lewis, L. F. 1993, 'Decision-aiding software for group decision making', in *Computer-aided decision analysis*, ed. Nagel, S., Quorum Books, Westport, Connecticut.

Liou, Y. I. & Chen, M. 1993, 'Using group support systems and joint application development for requirements specification', *Journal of management Information Systems,* vol. 10, no. 3, pp. 25-42.

Mayo, E. 1945, *The Socila Problem of an Industrial Civilisation*, Harvard University Press, Cambridge Mass.

McGregor, D. 1960, *The Human Side of Enterprise,* McGraw Hill, New York.

McGregor, D. 1969, *Theory X and theory Y: The work of Douglas McGregor*, BNA Films, Washington, D.C.

Pascale, R. 1990, *Managing on the Edge,* Simon & Schuster, New York.

Salzer-Morling 1998, 'As God Created theEarth... A Saga that Makes Sense?' in *Discourse + Organization,* eds. Grant, T., Keenoy, T. & Oswick, C., Sage, London.

Semler, R. 2000, 'How we went digital without a strategy', *Harvard Business Review,* vol. September/October, pp. 51-58.

Senge, P. M. 1992, 'Mental Models', *Planning REview,* vol. 20, no. 2, p. 5.

Stacey, R. D., Griffin, D. & Shaw, P. 2000, *Complexity and Management: Fad or radical challenge to systems thinking,* Routledge, London.

Tapscott, D. & Caston, A. 1993, *Paradigm Shift: The new promise of Information Technology,* McGraw-Hill, New York.

Taylor, F. W. 1911, *The Principles of Scientific Management*, Harper Row, New York.

Trist, E. 1981, 'The Evolution of Sociotechnical Systems', eds. Van. de Ven, A. H. & Joyce, W. F., Wiley Interscience, New York.

Van de Ven, A. & Polley, D. 1992, 'Learning While Innovating', *Organisational Science,* vol. 3, no. 1, pp. 92-115.

Whiteley, A. M. & McCabe, M. M. 2001, *Sustainable Change:A case at the waterfront*, Vineyard Press, Perth.

Whiteley, A. M. & Wood, E. 2000, *Decision Making and Decision Tools: GSS and Complex Adaptive Systems Theory (Working Paper 00.13)*, Curtin University, Graduate School of Business, Perth, Western Australia.

Zohar, D. 1997, *Rewriting the Corporate Brain:Using New Science to Rethink How We Structure ans Lead Organisations*, Berrett-Koehler, San Francisco.

Other Titles in the Graduate School of Business Working Paper Series (Back page title)

94.01 Nowak, M and Whiteley, A, New Strategies to Develop ‘Real’ Engagement by Students with the Learning of the Knowledge and Skills Required for International Competitiveness.

95.01 Travaglione, T, Pinto, D and Cacioppe, R, Employee Share Plans: Recent Industrial Relations and Taxation Developments.

95.02 Whiteley, A, Levantard, M and Anderson, P, Critical Thinking/Questioning Skills a Meta-Model for Cognitive Development in Management Education.

95.03 Cacioppe, R and Bedford, R, Evaluation of a Participative Change Program Over Two Years.

95.04 Travaglione, T and Williams, P, The Move to Individualism by The Western Australian Education Department.

95.05 Firns, I, Travaglione, A, and Fink, S, Employee Commitment in Westrail.

95.06 Clifford, P and Evans, R, Corporate Governance: A Question of Independence

95.07 Silcox, S, Cacioppe, R and Soutar, G, Quality Sub-Cultures and Their Influence on Change Interventions

96.01 Whiteley, A, McCabe, M, and Savery, L. Preliminary Findings on Waterside Workers’ Responses to the role of Unions following the 1991 Enterprise Based Agreement.

96.02 Whiteley, A, McCabe, M, and Savery, L. Human Resources on the Waterfront: Managing History

96.03 Whiteley, A. Report on a preliminary investigation into the need and support for a (tentatively named) director’s development centre

97.01 Whiteley, A, McCabe, M, and Savery, L. Trust and Communication Development Needs: An Australian Waterfront Study

97.03 Cacioppe, R. An Integrated Model and Approach for the Design of Effective Leadership Development Programs

97.04 McCabe, M. & Nowak, M. The Impact of Baby Boomers’ Longevity: Are There Implications for Boards of Directors

98.01 Whiteley, A., McCabe, M., Buoy, L., Howie, F., Klass, D., Latham, J., Bickley, M. & Luckeenarain, L. Planning the Qualitative Research Interview.

98.02 Nowak, M. J. The Role of Information in Internal Labour Markets.

99.01 Stockwell, C. & Nowak, M. An Application of National Competition Policy: Parking Services, City of Perth.

00.01 Dumont, E. & Cacioppe, R. Leadership Directions for Non-Profit Boards of Management

00.02 Whiteley, A. Grounded Research: A Modified Grounded Theory for the Business Setting

00.03 Whiteley, A. & Wood, E. Decision Making and Decision Taking: GSS and Complex Adaptive Systems Theory

01.01 Niki Lukviarman. Key Characteristics of Corporate Governance: The Case of Indonesia.

01.02 Nowak, M.J. & McCabe, M. Information Costs and the Role of the Independent Corporate Director

01.03 Galvin, P. Product modularity and the contextual factors that determine its use as a strategic tool

02.01 Whiteley, A. Rigour in Qualitative Research.

02.02 Galvin, P., Davies, J. & Simons, R. An Exploratory Investigation into Strategic Groups and Strategic Blocks as an Explanation for Patterns of Rivalry in the International Airline Industry.

02.03 Galvin, P. & Rice, J. The Role of Modularity in Knowledge Protection and Diffusion: The Case of Nokia and Ericsson

02.04 Galvin, P. & Rice, J. Alliance Dynamics in Response to Declining Environmental Munificence: The Case of Bluetooth.

02.05 McCabe, M. & Nowak, M. An Emergent Model of Best Practice in Corporate Governance

03.01 Wood, E. & Blyth, D. Using Group Support Systems Technology to Develop Comprehensive Risk Profiles on Major Resource Projects: A Case Study and Process Template

For further information on this discussion paper series please contact:

Ms Judy Gaebler, Secretary to the Director
Graduate School of Business, Curtin Business School
Curtin University of Technology
GPO Box U1987, Perth, Western Australia 6001

Fax: 61 8 9266 3368   Telephone: 61 8 9266 3366