Goal-Framing and Breakdowns in the Design of Boundary-Spanning Information Systems[1]

Susan Gasson
the iSchool at Drexel,
Drexel University,
Philadelphia, USA
***email*: sgasson@ischool.drexel.edu**

Abstract

Design groups which span organizational boundaries are becoming increasingly common and present a unique problem for information system design. Such groups attempt to integrate design understanding across many different domains of expertise and knowledge. This means that no one design group participant is likely to understand either the problem domain or the solution domain in full.

This paper critiques traditional and emerging process models of design, in terms of their applicability for boundary-spanning design groups. Following the use of the term in other fields such as architecture, "design" is viewed here as the process of conceptualizing, developing, refining, and evaluating a computer-based information systems solution to an organizational problem, rather than as a single stage of an IT systems development life-cycle. Findings from studies of boundary-spanning design groups were synthesized across multiple levels of analysis to present an integrated model of design for such groups. The model builds on the notion of technological frames, to develop a theory of framing at the three levels of project coordination, group consensus-building, and individual learning and dissonance. The model is discussed in terms of how this would operate as a spiral model. Implications for theory and practice are discussed.

Keywords: Information systems design, wicked problems, cognitive breakdowns, distributed cognition, boundary-spanning design.

1. Introduction

Organizations are increasingly focusing on the design of organizational information systems that incorporate both business process and information technology (IT) change, to achieve alignment between business and IT objectives. These information systems (IS) span functional boundaries and business units, requiring many different disciplinary, organizational and political interests to be negotiated and multiple, diverse ways of working to be reconciled. Organizational IS design, especially when this spans functional or business unit boundaries, conceptualizes and implements changes to the organization and practices of work, as well as technological change. IS managers thus need to balance the need for actionable, objective definitions of IT-related change with a need for radical, boundary-spanning inquiry into what needs to change (Engeström et al., 1995; Liu et al., 2002). In the organizational and MIS literature, we have an increasing awareness of the emergent nature of IT-systems definition and its embeddedness within wider processes of organizational problem-solving and business-process change (Bergman et al., 2002a; Markus et al., 2002). The term *information system* is used here to mean an integrated social system of organizational actors, using information to perform purposeful activity, who

[1] This material is based upon work supported by the National Science Foundation under Grant No. IIS-0347595. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

may or may not use computer-based technology to facilitate their work and to provide information (Checkland (Checkland, 1981; Hirschheim, 1986; Land, 1992). Following the use of the term in other fields such as architecture, product design, or graphic design, the term *design* is viewed here as the complete process of conceptualizing, developing, refining, and evaluating an artifact-related solution to a problem, rather than as a single stage of the IT systems development life-cycle (Lawson, 2000). The critical processes of boundary-spanning design take place "upstream" of the typical system development lifecycle, when organizational change and organizational problem-solving provide "early requirements" for an IT system and requirements for business process change. By definition, however, an integrated view of design also includes systems development and business process change, as design in boundary-spanning contexts is emergent and so is not complete until a solution has been evaluated and refined. This form of design involves emergent knowledge processes, for which goals emerge during the process of design (Markus et al., 2002). It involves a dialectic between the processes of inquiry and processes of closure, whereas the typical IS design method focuses exclusively on closure.

2. Competing Models of The IS Design Process

2.1 Hierarchical Decomposition

The dominant process model underlying the design science literatures is Simon's (1973) semi-rational model of ill-structured problem-solving. IS design is associated with a consensus problem definition and a set of goals for its resolution *that are defined by structures inherent in the situation* (Simon, 1973; Simon, 1996). Because human beings cannot cope sufficiently with complexity, analysts unconsciously bound and select the set of variables, constraints, and goals that they consider, to produce a "satisficing" solution. A set of consensual goals for change is agreed, then the analyst engages in a process of problem structuring and decompositional means-end analysis – a "gap analysis" between the current situation (the problem-space), and the desired situation (the solution-space[i]). The solution space (that set of potential solution elements available to the analyst) is explored through hierarchical decomposition of the structures inherent in the problem space. This process is shown in Figure 1.

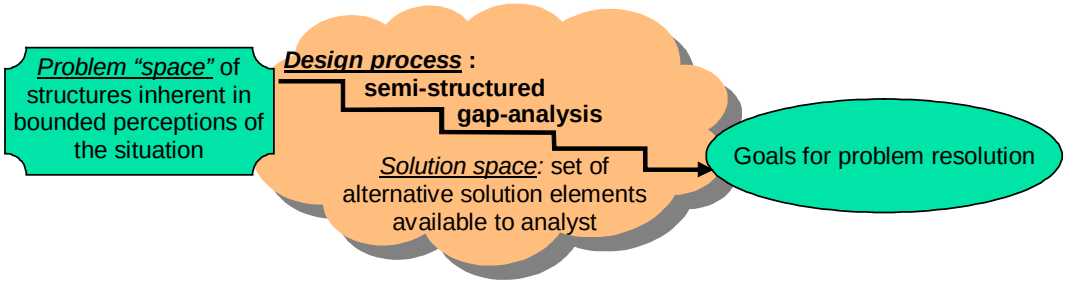


Figure 1: A Traditional Model of the IS Design Process
(derived from Simon, 1973; Simon, 1996)

Simon's model of design underlies the waterfall model, which is still used to manage the majority of IS design and development projects (Barry and Lang, 2003). The most significant problem with this model is that it assumes that the structure of the problem space is given, i.e. *inherent in the situation*. While designers may simplify and bound the problem-structure, to produce a problem-definition that is capable of analysis, this is viewed as a reduction, rather than a redefinition of the problem (Simon, 1996). The stopping-point for design from this perspective is thus when a set of solution criteria based on this problem structure are satisfied (Simon, 1973; Simon, 1996). The majority of design approaches are fundamentally flawed, as they apply Simon's goal-directed model to situations where the problem-structure (and hence the goal(s) for

change) is subject to interpretation and negotiation (Dorst, 2006). This approach may be appropriate for the design of a technology artifact, whose role and purpose is well understood. But the human, social, and organizational contingencies of design are easily subverted by activities within the SDLC, as the emphasis is on managing methods (consistency of process) rather than process (fit of design outcomes with context) (Greenbaum and Kyng, 1992; Mumford, 2003; Scarbrough and Corbett, 1991). The paradigm of means-end analysis grants problems an objective existence, ignoring the need for problem negotiation (Pries-Heje and Baskerville, 2006; Winograd and Flores, 1986). But this is only possible because the traditional model of design ignores the emergent nature of organizational goals, focusing instead of artifact design goals.

2.2 Opportunistic Design

In contrast, design goals for IS that span organizational boundaries evolve as the group of design participants and stakeholders engage with competing and partial understandings of the design problem. Goals and plans do not tend to provide a guide for the action that follows (Mintzberg and Waters, 1985; Suchman, 1987). A statement of goals most frequently reflects *post hoc* reasoning about the meanings and problem-structures perceived in the situation after change has been implemented (Lave, 1988; Nardi, 1995). Actors engaged in boundary-spanning design redefine the problem periodically, based on reflective, experiential learning. This changes both the design goals and the locus of change (Brown and Duguid, 1992; Truex et al., 1999; Valkenburg and Dorst, 1998). Design participants and stakeholders face waves of "wicked problems", that are unique, novel, subjective, only partially understood, highly interrelated, and involve a negotiated, evolving problem-definition. No objective criteria for a solution may be formulated for a wicked problem – it can be resolved only through argumentation, until stakeholder needs are minimally satisfied. Where tame (traditionally-conceived) design problems can be defined, analyzed, and decomposed to arrive at a solution, wicked problems require evolutionary design approaches, based on argumentation, emergent problem representation, and goal-negotiation across stakeholder groups (Buckingham-Shum et al., 1997; Conklin and Weil, 1997; Fischer and Ostwald, 2005; Rittel, 1972; Rittel and Webber, 1973). Organizational problems are situated within a specific social context, culture, and a set of work-practices which require knowledge that is local, often implicit, and certainly emergent at the generic level of design that spans organizational boundaries (Brown and Duguid, 1992; Lave and Wenger, 1991; Weick, 1995). Goals emerge through reflective interactions with the social context of the problem and through designer-stakeholder dialog (Boland and Tenkasi, 1995; Carlile, 2002; Markus et al., 2002). Designers are "thrown" into a situation where local actors understand and assume problem-elements, solution constraints, and criteria for a solution that are embedded in the experience of the organization and its evolving goals. They must engage with local practices and meanings, to acquire an understanding of the generic subjectivity that is shared by other design participants and stakeholders with whom they interact. It is with this understanding that we seek an alternative model of the design process – one that fits with the reframing of problems and solution-search that appears to underlie emergent knowledge processes. The assumption that problem-structures are taken-as-given is common in technocentric or artifact-focused analyses of design processes, as it fits well with the definition of design objects, behaviors, and functions (e.g. Cooper and Reimann, 2003; Gero et al., 1992). But empirical evidence from software engineering design suggests that experienced IS designers appear to be "opportunistic" in their use of contextual information to identify problem-structures, based on partial solutions available to them (Ball and Ormerod, 1995; Guindon, 1990). Analysts extrapolate solutions from problems encountered previously, incorporating implicit knowledge and implied requirements into the

framing of new solutions. If there are no solutions available for a design problem, as currently defined, *the problem may be redefined to fit available solutions* (Dorst and Dijkhuis, 1995; Malhotra et al., 1980; Turner, 1987; Urquhart, 2001). Such findings have led some authors to suggest that design may be a-methodical (Truex et al., 2000). But that does not mean that design in such projects lacks a coherent process. From the evidence presented on opportunistic design, the answer appears to lie in a model that parallels the "garbage can" theory of problem-solving (Cohen et al., 1972). Perhaps designers do not analyze the problem space at all, but just identify plausible solution fragments and fit these to an evolving problem-space bricolage?

2.3 Convergent Design

Halfway between the ordered universe of hierarchical decomposition and the opportunistic anarchy of the garbage can model, lies a third model of design. This model, originating from studies of software program design views design as the co-evolution of a problem-space and a solution space (Maher and Poon, 1996; Maher and Tang, 2003). A similar process has also been observed in think-aloud protocol studies of creative product design (Dorst and Cross, 2001) and is similar to the concept of "grinding" between problem and solution space (Purao et al., 2003). But a co-evolution process model derived from studies of individual cognition in software design may be overly-simplistic for group collaboration, especially when this involves stakeholders from multiple backgrounds and functional disciplines. The concept of solution space evolution is developed further by Bergman et al (2002a & 2002b (Bergman et al., 2002a), to distinguish between the solution-space as it exists now (i.e. the ecology of functional requirements that solves current perceptions of the problem-structure) and a potential, future solution-structure that results from analyst deliberation. But this perspective treats the problem-space as *negotiated* rather than evolutionary: it does not deal with the evolution of perceived problem-structures discussed in the previous section. Experienced analysts are subjective in their definition of the problem-space and actively recognize learning outcomes as they proceed with a design. Individuals develop a richer analysis of the problem-space as they proceed with a design; this process is multiplied in group design and so breakdowns in group consensus occur, during which the group must reassess what they know and what they can assemble as a group (Gasson, 2006; Gasson, 2007).

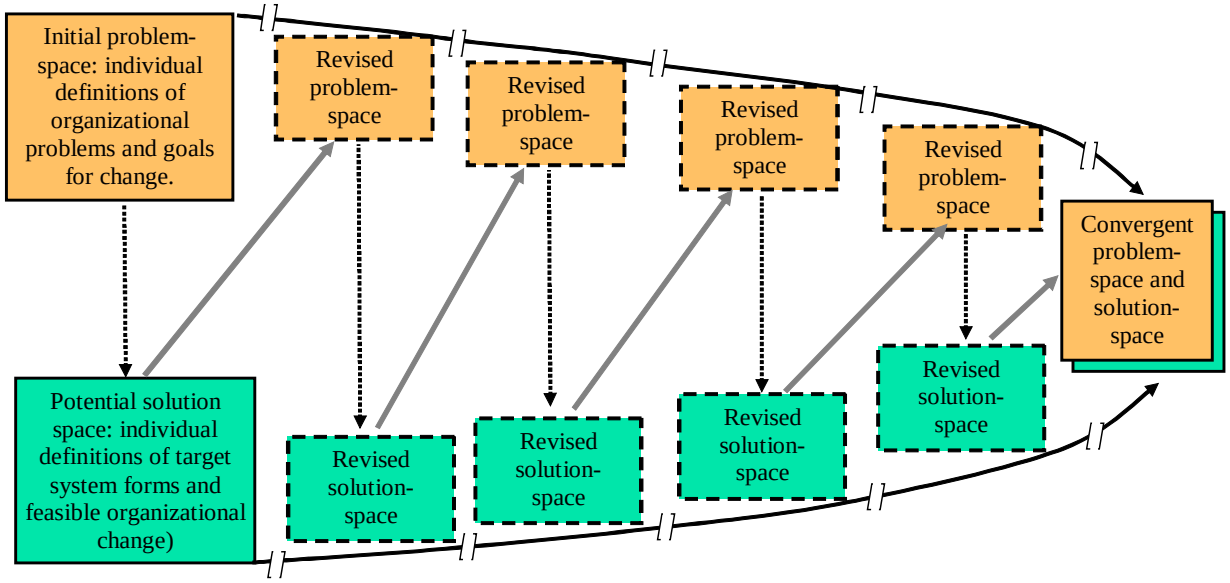


Figure 2: Design as Punctuated Co-Evolution Between Problem Space and Solution Space

The process by which groups develop a solution in situated studies of collaborative problem-solving and information system design has been described as "punctuated equilibrium". In this process, aligned problem-perspectives are punctuated by disruptions that cause the group to completely redefine their problem and as a consequence, rethink their solution (Gersick, 1988; Gersick, 1991). Gersick (1991) hypothesizes that this is driven by group perceptions of time-constraints, observing that there is always a major redefinition of the problem structure midway through the project duration. But other studies have observed more frequent disruptions to group consensus. A similar type of punctuated or interrupted process has been observed in user-analyst system requirements definition (Newman and Robey, 1992) and in larger scale studies of heterogeneous IS requirements analysis (Bergman et al., 2002a; Bergman et al., 2002b). A suggested model, reconciling the co-evolution process with a punctuated equilibrium process, is given in Figure 2. If we accept this model as a working theory of design, to be explored through empirical research, the question arises of how design operates in a boundary-spanning design group. This requires an analytical lens that explores not only the evolution of problem and solution spaces over time, but also the catalysts that drive an evolving design consensus across design participants and stakeholders from multiple knowledge domains. To investigate these issues, I employ the theoretical lens of distributed cognition in framing design.

2.4 Distributed Cognition and Breakdowns In Design

Most studies of design groups simply assume that individual understanding – how the design problem and solution are framed – simply scale up to group design. Shared understanding and knowledge are reflected in a shared group culture; the absence of a shared culture and shared practices often indicates diversity in how group members understand what needs to be done (Levine and Moreland, 1999). But group relations change when conditions and deliverables change. Groups faced with dynamic conditions must develop interdependence mechanisms: coordinating their activities, developing trust, and coming to rely on each other. An IS design problem situation may be perceived as a wicked problem simply because of repeated breakdowns in trust between group members. Negotiating trust relationships may require methods that do not directly challenge individual assumptions and frameworks of interpretation.

Applying a theory of distributed cognition allows us to view understanding as *stretched across*, rather than shared between collaborating individuals from different functional groups (Lave, 1988; Star, 1989). To negotiate the Venn diagram of individual understandings that results, shared work-practices are mediated via a web of well-defined functional roles that permit distinct framing perspectives to be reconciled only when this is necessary for mutual task or outcome coordination (Boland et al., 1994; Hollan et al., 2002; Hutchins, 1995; Star, 1989).. This is especially pertinent to IS design projects involving boundary-spanning groups of participants, but its significance is also demonstrated more generally by Faraj and Sproull, who performed a large-scale survey of IS design groups and found that the management of distributed cognition was paramount for IS success. A knowledge of *who-knows-what* across a distributed set of participants was far more important to success than the use of specific methodologies or techniques (Faraj and Sproull, 2000). Homogeneity may be counterproductive in producing an effective design (Flor and Hutchins, 1991). Innovation requires a *collective requisite variety* (Weick, 1987) of design frames:

> "When technical systems have more variety than a single individual can comprehend, one of the few ways humans can match this variety is by networks and teams of divergent individuals. ... If people look for different things, when their observations are pooled, they collectively see more than any one of them alone would see." *(Weick, 1987, page 334).*

But while variety is critical to an effective design, some form of consensus in the adoption of shared frames is critical to design closure. Information system designers, "successful in sharing plans and goals, create an environment in which efficient communication can occur" (Flor and Hutchins, 1991). The notion of frame congruence is employed to describe a state of group consensus, where individual framing perspectives are not identical, but are sufficiently shared to form the basis for that efficient communication. Sociological studies of IS have adopted the concept of a *technological frame* to provide a structure for jointly-internalized beliefs about a situation. A technological frame cannot be viewed as pertaining to an individual or a group, but is seen as "located between actors" (Bijker 1987). It denotes a shared construct *to the group*, that they call upon to guide their joint actions, but does not indicate a *totally* shared understanding of the domain to which the frame relates. A technological frame corresponds to a form of implicit consensus, that represents the intersections of individual group members' frames of reference. The construct is therefore relevant to the analysis of interactions between interest groups, or to understand how members of a specific group define the role and purpose of a technology in respect of their organizational situation (Orlikowski and Gash 1994). It is clear that consensus does not imply totally shared beliefs and interpretations (Fiol 1994), but merely accommodates others' perspectives when these do not entirely conflict with one's own. Technological frame congruence occurs when frames are related in structure (possessing common categories of frames) and content (with similar values in the common categories) (Orlikowski and Gash, 1994).

Because each context of analysis brings different, domain-related technological frames to bear on organizational problem-solving or design, there is little guidance on how to analyze technological frames. Orlikowski and Gash (1994) suggest that structure and content (defined above) may be considered as elements of frame congruence among members of a group participating in similar tasks with a shared culture, but suggest that frame analysis should be related to the domains of situation at hand. Davidson (2002; 2006) echoes this view, arguing that the analysis of technological frames requires researchers to elicit subjects' deeply-held assumptions and knowledge about a specific IT application in its organizational context. The framing perspectives applied by an individual are communicated through the metaphors and analogies that they employ, which in turn generate technological frames that reflect the shared organizational practices, cultures, and genres in which group members participate (Davidson, 2002; Lin and Silva, 2005).

Frames may be mediated and negotiated across knowledge-domain boundaries by the use of boundary-objects (Star, 1989). These are representations or cognitive artifacts that are sufficiently well-defined to support collaboration across knowledge-domain boundaries, but sufficiently elastic to permit elements of the "shared understanding" that ensues to be interpreted differently by various individuals in the design group. Different types of boundary object support different types of shared understanding. *Repositories* allow collaborating individuals to agree on the elements necessary for a solution, but to employ different domain-specific categorizations of those elements. *Standardized forms, methods, and procedures* enforce normative work practices across domains and provide a shared format for solving problems. *Models and ideal-type objects* provide an abstraction that works for all knowledge domains. *Terrain with coincident boundaries (maps)* provide common boundaries, or scope of analysis, while permitting different conceptualizations of the elements within that scope. Boundary objects thus allow collaboration around specific tasks without the need for a completely shared understanding (Carlile, 2004; Star, 1989).

The use of specific external representations (e.g. diagrammatic models) may indicate a distributed frame alignment (Flor and Hutchins, 1991), or alternatively may surface *breakdowns* around group design consensus (Winograd and Flores, 1986). The centrality of breakdown in design is related to Heidegger's (1962) argument that objects and properties are not inherent in the world, but are simply "ready-to-hand": unreflectively incorporated into automatic routines. A disconnect in the seamlessness of their use produces a breakdown. The artifact becomes visible in use, or "present-at-hand": the subject of reflection and redefinition (Winograd and Flores, 1986). Automaticity in use results from the individual's experience of incorporating similar artifacts into one's work-routines. When I am hammering in a nail I do not concern myself with the process of hammering, I just do it. When the hammer is too heavy for the nail or if the hammer breaks, I am forced to reflect not only on how the hammer is constructed, but also on the process of hammering-in-context – how a hammer is used, in what circumstances certain hammer designs work, and why certain hammer designs fit with different types of hammering-process. As a result, I understand both hammer design and the process of hammering much more deeply, permitting me to select the right hammer for the job.

So breakdowns may be productive in design. They force a disconnect in the assumptional flow, leading to a reexamination of design rationale and objectives. Extending the concept to group processes, it is possible that *collective breakdowns* hold the key to achieving requisite variety in boundary-spanning design. An early homogenization of problem perspectives can be harmful, resulting in premature specification – locking in the form and functions of the solution before the requirements for change have been understood. But premature specification occurs even when the method in use *should* sensitize group members to the multiple perspectives required for effective problem mediation. Groups of IS professionals trained in methods for rapid problem closure are highly homogeneous in their culture and focus. Yet that culture and focus may need to be disrupted if IS professionals are to engage with functional managers and stakeholders in boundary-spanning IS design.

3. A Socially-Situated Model of Punctuated Emergence In Design

This model is synthesized from findings from two studies of distributed groups, responsible for the co-design of business processes and their supporting IT systems. The outputs perceived by both studies were framed in terms of requirements for change and high-level IT system requirements. In that sense, the studies could be likened to early requirements analysis for IT systems, except that each of the groups – both of which included experienced and competent IS professionals – clearly distinguished their process from IT systems development. The membership of both groups had been selected to involve people with extensive experience and success in determining organizational or IS change. Both groups referred to their process as design and clearly produced conceptualizations that gave a new *form* to the objects of design, whether artifacts or processes. The suggested model is shown in Figure 3.

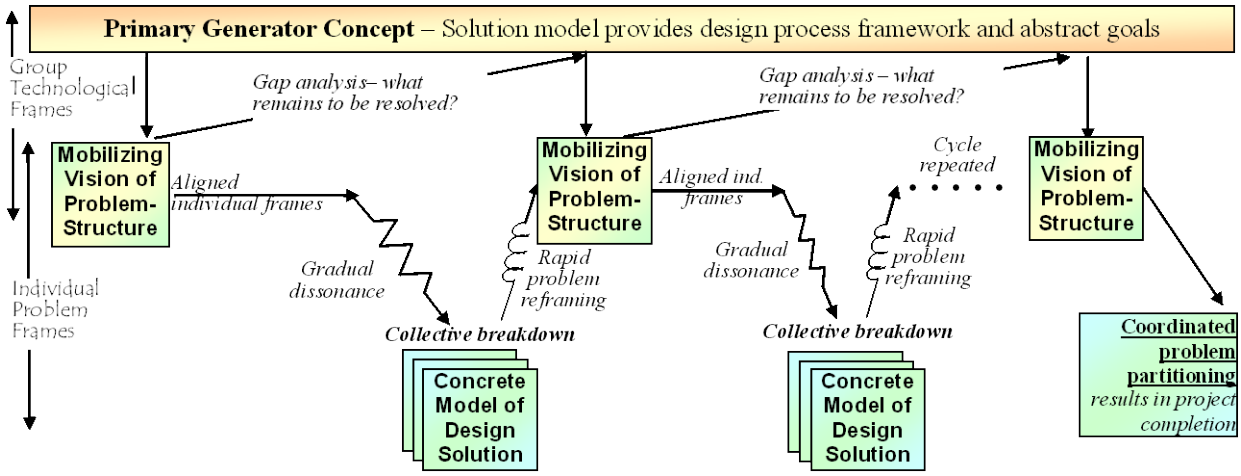


Figure 3: Multilevel Goals and Spaces In Design

Figure three contains three levels of design space. The *Level 1 design space* is a set of abstract goals that is defined through a primary generator concept for the design (Darke, 1979). Darke observed that architectural design groups rapidly generated a suitable solution structure, such as "we thought we would build a traditional ranch house," that ensured that everyone was on the same page for the design. The primary generator provided an overall structure for the solution-space that, by inference, defined the problem-space as it is originally framed by the design group. For example, in one of the studies, the design group defined the primary generator concept as a six-stage, linear process to respond to customer invitations to bid for a contract. In another, the group defined the primary generator concept as a Financial Information System to register students for courses and evaluate their eligibility for Financial Aid. In both cases, the primary generator concept provided a unifying structure for the design *process*, rather than a persistent problem-structure. In the first case, the eventual problem-structure reflected in group design documentation, had many more than six stages and many iterations between and within stages. Yet the group still referred to the six-stage process, as this allowed them to divide the intellectual labor of design between group members and to estimate progress against their initially-conceived, abstract design goals of improving IS support for activities at each of the original six stages. In the second case, the design group added many more areas of functionality, integrating support for the introduction of new programs and flexible delivery schedule management into the system. Yet again, they described their overall goals in terms of the original structure, estimating progress in design against this definition.

The *Level 2 design space* is defined by a series of mobilizing visions, each of which developed and built on the previous one. The mobilizing vision reflects a more detailed framework for the solution space, as understood by the group as a whole. In the first study, the design team defined first that they would build an electronic document repository, then a virtual team support-space, and then coordination support for a customer service process that aligned with the strategic planning process, and so on. In the second study, the design team defined that they would redefine their financial planning system, then that they would define a parallel information system to capture "floating" students, then that they would define an integrated enterprise system that registered the two types of student and validated financial aid requirements, and so on. In both projects, as individual understanding of the design progressed, each vision proved inadequate for the purposes of group consensus. There would be a rapid period of dissonance, as individuals disagreed with how to implement the detailed design around the current mobilizing vision. This would result in a collective breakdown, when the group

decided that the current mobilizing vision was inadequate. It would be replaced rapidly by a substitute, through a brainstorming process that developed ideas and integrated individuals' perceptions of lacunae into a richer mobilizing vision.

The *Level 3 design space* is that set of individual problem frames, that remains unarticulated but provides constant validation for both the mobilizing vision and the representational models used to achieve consensus in design. I have been surprised, over a series of studies, at how accommodating design group members are, in absorbing obvious differences between their own problem-frames and those of other group members. It appears that learning is seen as an explicit objective of the process at the individual level, and this only results in severe dissonance when an external artifact must be produced, that represents group consensus on the solution-space. For example, the first group had to produce a presentation to senior management, on which a go/no-go decision for the design would be based. The second group had to produce an interim specification for changes to the financial management system. In both cases, the fragile group consensus on what needed to change was disrupted by the cumulative dissonance between individual mental models of the problem-space and the collective target solution model required for this output. The group suffered a collective breakdown, during which the mobilizing vision was rapidly replaced, as described above, to refine the collective problem scope and structure.

What is interesting is how the stopping-point was reached. These are boundary-spanning design initiatives. They spanned many different functions of the organization: managers and analysts from many different organizational roles were involved. A consensus design was not possible, as each participant in the process understood the organizational business processes partially and from the perspective of their stakeholder group interests. In each case, the design group appeared to reach a consensus that they could implement the complex changes to IT systems and business processes that they had defined at a high level, by means of a division of labor. Group members who were considered knowledgeable in one or more areas were delegated to oversee the implementation of design changes to business processes or IT systems, in sub-systems of the design. Each group took a jigsaw-puzzle approach, breaking design implementation down into a set of component parts that matched group members' areas of expertise. In many cases, the implementation also required clarifying a detailed design for that sub-component. In both cases, the stopping point did not appear to be collective understanding: it was clear that none of the design group members understood the design as a whole. Instead, the stopping-point appeared to be reached when the group felt that it had achieved sufficient levels of *shared trust* around the way in which the primary generator concept was interpreted. This seemed to depend on how others framed the abstract goals of the design, when jointly producing a collective solution model. In both groups, IT system development was inextricably intertwined with business process experimentation, resulting in IT system prototyping on an ongoing basis as new design requirements emerged. The final implementation and roll-out of IT system components tended to be performed under the leadership of a technical member of the design team, but was evaluated on an ongoing basis by other members of the team, in terms of the effectiveness of IT support for the newly-designed business processes.

4. Discussion: Implications of the Model

The question remains of how far the punctuated equilibrium model shown in Figure 2 represents the empirically observed model of design for complex, boundary-spanning design contexts. The model shown in Figure 3 combines elements of a co-evolution model with elements of a punctuated equilibrium model of organizational change. But it adds an additional

layer of complexity, distinguishing between the primary generator concept (Darke, 1979), that frames the design process and the overarching solution-space, from the high-level system vision and objectives that frame the evolving problem-space and its implications for design objectives and the target solution space. This model may be unique to boundary-spanning design projects and the projects involving the co-design of business processes and IT systems. In these change projects, the problem-space is so complex that no individual can completely comprehend it. This leads to the process of evolving consensus, punctuated by collective breakdowns, that was described above. The eventual designed solution space consists of a bricolage (Ciborra, 2002) – an ecology of partial solution-spaces advocated by individual design group members. From these findings, it appears to be an evolving problem-space that drives design and not an evolving solution-space. High-level change goals and visions refer to an emergent realization of the collective problem-structure and not to an understanding of the detailed, technical logic (as is the case for less complex IT design).

An organizational IS design proceeds as group members suggest elements that fuse into a collective framing of the organizational problem (the mobilizing vision). It progresses as individual members test this framing against implicitly held partial solutions and fail to find a good fit, triggering a collective breakdown that leads to the shared mobilizing vision being reframed and replaced. Closure ensues when individuals have developed sufficient confidence that other members of the group will carry this problem-frame with them as they implement partial technical and process-change solutions. But new areas of design inquiry may open up as another closes. It appears to be very difficult to estimate design progress in these design projects, simply because no one individual understands the solution-space, or the problem-space in its entirety. Design progress is achieved through explicit learning on the part of individuals, maintenance of a fragile consensus, and acceptance of repeated conflicts as the consensus breaks down. Design completion is achieved through construction of a shared trust, that recognizes that – no matter how different one individual's solution-space appears to be from another's – the solution-space advocated by others complements that individual's solution-space and will enable them collectively to achieve the abstract set of design goals that they all share.

This results in a very different form of "spiral" model to the ones that are usually advocated, as shown in Figure 4. To ensure progress in design, it appears that a series of regular deliverables must be planned, that require the production of a shared rationale for the target design, for an influential senior manage. The latter element is significant because of the cost of consensus. If there is a high cost of change at a later point, as there is when the expectations of an influential senior manager are raised, group members cannot ignore differences in framing the organizational problem, as they need to "get it right." This produces a collective breakdown, which leads to a refined mobilizing vision, as the group reframes their collective problem-structure. The key deliverable of each major cycle would therefore appear to be a reconceptualized model of the problem-structure, rather than a refined model of the solution-space.

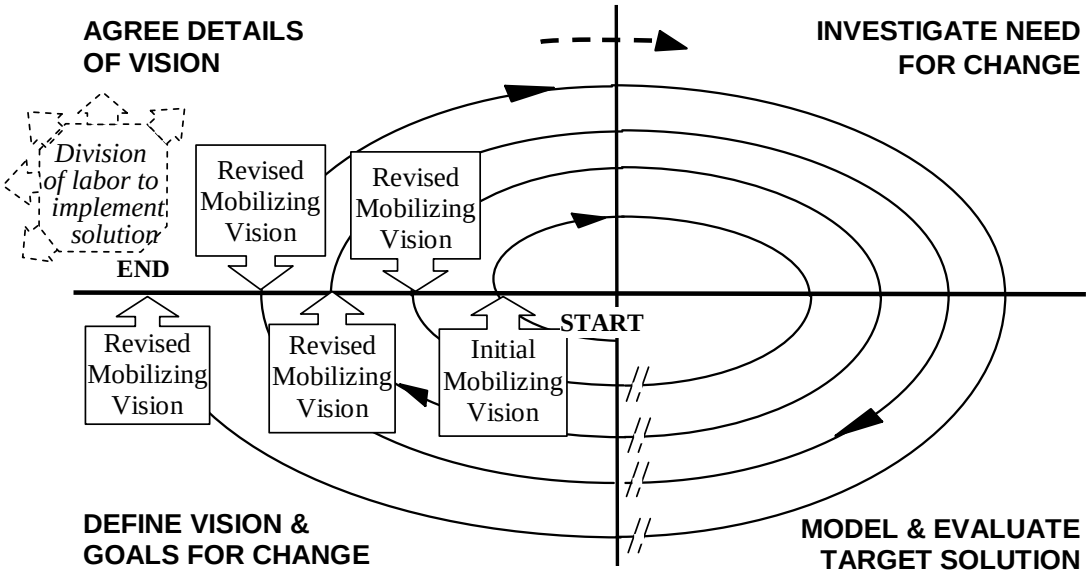
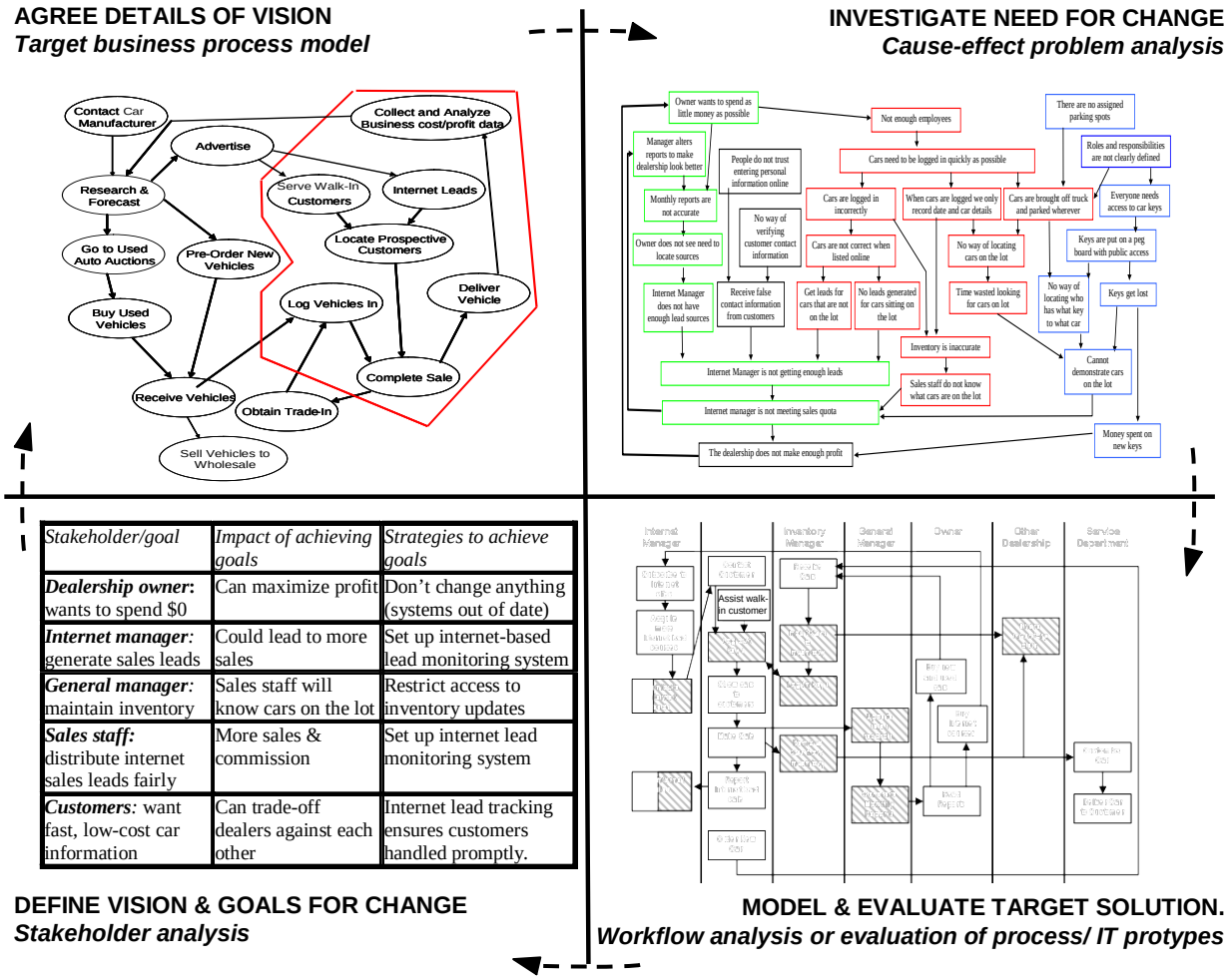


Figure 4: Revised Spiral Model To Support Problem-Directed Design

A collective breakdown is likely to occur as the group defines its target solution for presentation to senior management. This forces the group to agree on a detailed, consensus model of process and system changes, with a high cost attached to failure (i.e. disappointed expectations on the part of senior management). It is at this stage that differences in individual problem-space frames surface, leading to a collective breakdown and a reframing of the shared problem-structure (the mobilizing vision). In IT systems design, problems are reduced to a complexity that can be easily modeled and agreed upon. But in organizational IS-related change projects, the co-design of business process systems and IT support for a set of emergent knowledge processes means that consensus is highly problematic. So the type of conflict considered counter-productive in IT systems design may well be highly productive and a necessary part of boundary-spanning systems design. Because of this, conflict needs to be channeled into revising the mobilizing vision, rather than suppressed or treated as undesirable.



Stakeholder/goal	*Impact of achieving goals*	*Strategies to achieve goals*
***Dealership owner*:** wants to spend $0	Can maximize profit	Don't change anything (systems out of date)
***Internet manager*:** generate sales leads	Could lead to more sales	Set up internet-based lead monitoring system
***General manager*:** maintain inventory	Sales staff will know cars on the lot	Restrict access to inventory updates
Sales staff: distribute internet sales leads fairly	More sales & commission	Set up internet lead monitoring system
***Customers*:** want fast, low-cost car information	Can trade-off dealers against each other	Internet lead tracking ensures customers handled promptly.

Figure 5: *Implementation* of Revised Spiral Model To Support Problem-Directed Design

This realization has led to experimentation with techniques for each stage of the spiral model. The framework shown in Figure 5 provides a putative method based on the findings reported above. If the design space is seen as a system of distributed cognition (the theoretical lens employed to obtain these findings), individual design spaces may be mediated by various boundary-objects, according to the type of "translation" required across knowledge domains and the degree to which shared understandings are kept intentionally vague (Carlile, 2004; Star, 1989). As discussed earlier, boundary objects take the form of representations, artifacts, and processes that are sufficiently well-defined to support collaboration across knowledge-domain boundaries, but sufficiently elastic to permit elements of the "shared understanding" that ensues to be interpreted differently by various individuals. The various stages of the spiral model may be supported with boundary-object representations as follows:

Define vision & goals for change – *perform a stakeholder analysis*. By compiling multiple definitions of goals for change via a stakeholder analysis, this part of the cycle permits the collection of a diverse set of goals, that may be interpreted and categorized differently by various stakeholders, without being reconciled in detail. In the first cycle, this representational method provides the basis for a shared *primary generator concept* (Darke, 1979), that provides a unifying concept which permits team coordination around a richly-visualized, high-level goal. In the first and subsequent cycles, a *mobilizing vision* is generated, that represents the situated problem-structure faced by the team for that cycle of design. Both the high-level primary

generator and the mobilizing vision are models that work for any domain, following Star's (1989) ideal-type boundary object. Because of this, they represent a consensus, while remaining sufficiently vague to permit collaboration while avoiding individual dissonance at this stage.

Agree details of vision – *produce a target business process model*. This part of the cycle combines a high-level business process representation of the mobilizing vision with a definition of process boundaries related to the current problem-structure. It aids the design team in agreeing which elements of the organizational situation are included – or excluded – from the problem-space to be analyzed. The high-level representation produces a map boundary object (Star's "terrain with coincident boundaries"), that permits design participants from different knowledge domains to populate the problem-space according to their individual understanding, but agree areas of responsibility. It therefore enables collaboration around a vague problem-structure which will not restrict exploration of the design problem and solution-spaces.

Investigate need for change – *perform a cause-effect problem analysis*. This representational method employs a simplified form of cognitive mapping (Eden et al., 1992), to explore organizational and technical problem structures in more detail. The problem representation acts as a model boundary object: this permits stakeholders to assemble a composite understanding of their collective problem-structure, while permitting individual interpretation of detailed problem elements. Exploring linked networks of causality in the organizational problem space permits stakeholders to explore elements of the design ecology that would not normally be considered legitimate for more traditional design methods, such as the political and functional ecologies identified Bergman et al. (2007). Stakeholders can categorize various problem-clusters, combining a model boundary object with a map type boundary object (e.g. by using various colored boxes as shown in the upper-right quartile of Figure 5). This problem-representation also acts as a repository type boundary object, which allows problem elements to be shelved for later resolution while permitting each stakeholder to categorize problem elements as they choose, while negotiating their structure. This allows stakeholders to define a shared problem-structure, then agree not to "go there" for some elements, for reasons that they need not clarify to others.

Model & evaluate target solution – *perform a workflow analysis, or prototype process changes and systems*. The final segment of the cycle permits stakeholders to clarify and design solutions to the problems explored in the previous segment, the problem analysis. This part of the process creates a set of standardized forms, methods, and procedures (Star, 1989), that represent the mechanisms and functions of target processes and systems. In early cycles, a detailed workflow model can be employed, as shown in Figure 5. For later cycles, business process and IT systems prototypes may be evaluated by design-team members. At this level of detail, the use of shared formats and normative process definitions cause design participants to experience a cognitive breakdown (Winograd and Flores, 1986), when their individual model diverges from the model-in-use by the group. When individual breakdowns reach a critical mass of dissent from the agreed model, a collective breakdown occurs. At this point, the group moves to the next cycle of design, where they redefine the problem-structure (the mobilizing vision) to accommodate differences in perspective.

The representational techniques discussed in this section are heavily influenced by systemic inquiry approaches, such as Soft Systems Methodology (Checkland, 1981) and cognitive maps (Eden et al., 1992). Employing a systemic approach to inquiry enables connections to be explored between elements that may appear unrelated, avoids privileging one voice over others (which typically happens in the unitary design approach employed for IT systems), and enables tacitly-held understandings to be surfaced and explored. This stage of the research therefore builds on the systemic analysis literature, as the most appropriate approach to exploring wicked

problems. Studies to explore and refine these methods are ongoing. Initial findings support the core role of collective breakdowns in producing a revised mobilizing vision for the next cycle, but this may be limited by normative interpretations of a "relevant" process-model for IT design, that are employed by IS professionals. These issues have been discussed elsewhere (Gasson, 2007) and will be explored further.

5. Conclusions

It was argued that design groups which span organizational boundaries are becoming increasingly common and present a unique problem for information system design. Such groups attempt to integrate design understanding across many different domains of expertise and knowledge. This means that no one design group participant is likely to understand either the problem domain or the solution domain in full. This paper critiqued current models of design in these circumstances and suggested that we need an alternative model for boundary-spanning design. Findings from studies of boundary-spanning design groups were synthesized across multiple levels of analysis to present an integrated model of design. This model builds on the notion of technological frames, to develop a theory of framing at the three levels of project coordination, group consensus-building, and individual learning and dissonance. The resulting model presents the interior of design in complex situations, from the perspective of distributed cognition (Boland et al., 1994; Hollan et al., 2002; Hutchins, 1995; Star, 1989).

The findings would indicate that the design of boundary-spanning information systems is not goal-driven but *problem-driven*, necessitating the use of multiple methods to represent problem complexity sufficiently. Cycles of inquiry were balanced by cycles of problem closure, where requirements for process change and IT system support were defined and evaluated in practice. This model would seem to fit with a prototyping approach that combines experimentation with systems of organizational processes and information management, and the supporting system provided by an IT artifact. In practice, this was the approach employed in each of the studies reported here. This could be interpreted as a sign that the design participants were incompetent or did not understand organizational or information management problems. As an ex IT professional, I would argue that this was not the case. Both teams of managers impressed me with their professionalism and their ability to understand highly complex problems outside of their immediate area of expertise or responsibility. Rather, the nature of boundary-spanning organizational problems is that they involve distributed, emergent knowledge processes, which necessitate a different form of design process (Markus et al., 2002). This paper has explored the resulting design process from the perspective of distributed cognition. The findings suggest a model of design that is driven by collective breakdowns in shared understanding and routine problem reconceptualization. Suggestions for supporting this design process with a spiral model were explored.

It has implications for practice: it is possible that we can bring about more rapid learning and design progress by introducing triggers for breakdowns into the group process of consensus-building – for example, by employing a process that has regular reporting points or that incorporates changes in the form of design deliverables that require group agreement around the solution-space (Gasson, 2007). It also has implications for theory. By understanding the multiple levels at which design operates, we can conceptualize design processes for many different contexts and for groups with many different combinations of expertise.

References

Ball, L.J. and Ormerod, T.C. (1995) "Structured and opportunistic processing in design: A critical discussion," *Int. Journal of Human-Computer Interaction* (43:1), pp. 131-151.

Barry, C. and Lang, M. (2003) "A comparison of 'traditional' and multimedia information systems development practices," *Information and Software Technology* (45:4), pp. 217-227.

Bergman, M., King, J., and Lyytinen, K. (2002a) "Large scale requirements analysis revisited: The need for understanding the political ecology of requirements engineering," *Requirements Engineering Journal* (7), pp. 152-171.

Bergman, M., King, J., and Lyytinen, K. (2002b) "Large Scale Requirements Analysis As Heterogeneous Engineering," *Scandinavian Journal of Information Systems* (14), pp. 37-55.

Bergman, M., Lyytinen, K., and Mark, G. (2007) "Boundary Objects in Design: An Ecological View of Design Artifacts," *Journal of the Association for Information Systems (JAIS)* (8:11), pp. 546-568.

Boland, R.J. and Tenkasi, R.V. (1995) "Perspective Making and Perspective Taking in Communities of Knowing," *Organization Science* (6:4), pp. 350-372.

Boland, R.J., Tenkasi, R.V., and Te'eni, D. (1994) "Designing Information Technology to Support Distributed Cognition," *Organization Science* (5:3), pp. 456-475.

Brown, J.S. and Duguid, P. (1992) "Enacting Design for the Workplace," in: *Usability: Turning Technologies Into Tools*, P.S. Adler and T.A. Winograd (eds.). New York, NY: ACM Press, pp. 164-197.

Buckingham-Shum, S., MacLean, A., Bellotti, V., and Hammond, N. (1997) "Graphical Argumentation and Design Cognition," *Human-Computer Interaction* (12:3), pp. 267-300.

Carlile, P.R. (2002) "A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development," in: *Organization Science*. INFORMS: Institute for Operations Research, p. 442.

Carlile, P.R. (2004) "Transferring, Translating, and Transforming: An Integrative Framework for Managing Knowledge Across Boundaries," *Organization Science* (15:5), pp. 555-568.

Checkland, P. (1981) *Systems Thinking Systems Practice*. Chichester UK: John Wiley & Sons.

Ciborra, C.U. (2002) *The Labyrinths of Information: Challenging the Wisdom of Systems*. Oxford UK: Oxford University Press.

Cohen, M.D., March, J.G., and J.P. Olsen (1972) "A Garbage-Can Model of Organizational Choice," *Administrative Science Quarterly* (17:1-25).

Conklin, E.J. and Weil, W. (1997) "Wicked problems: Naming the pain in organizations." Retrieved April 5, 2005, from http://www.touchstone.com/tr/wp/wicked.html

Cooper, A. and Reimann, R. (2003) *About Face 2.0: The Essentials of Interaction Design*. Hoboken, NJ: Wiley.

Darke, J. (1979) "The Primary Generator And The Design Process," *Design Studies* (1:1), pp. Reprinted in N. Cross [Ed.] Developments In Design Methodology, 1984, J. Wiley & Sons Chichester, pp. 1175-1188.

Davidson, E.J. (2002) "Technology Frames and Framing: A Socio-Cognitive Investigation of Requirements Determination," *MIS Quarterly* (26:4), pp. 329-358.

Davidson, E.J. (2006) "A Technological Frames Perspective on Information Technology and Organizational Change," *The Journal of Applied Behavioral Science* (42:1), pp. 23-29.

Dorst, C.H. (2006) "Design Problems and Design Paradoxes," *Design Issues* (22:3), pp. 4-17.

Dorst, C.H. and Cross, N.G. (2001) "Creativity in the design process: co-evolution of problem–solution," *Design Studies* (22:5), pp. 425-437.

Dorst, K. and Dijkhuis, J. (1995) "Comparing paradigms for describing design activity," *Design Studies* (16:2), p 261.

Eden, C., Ackerman, F., and Cropper, S. (1992) "The analysis of cause maps," *Journal of Management Studies* (29), pp. 309–323.

Engeström, Y., Engeström, R., and Karkkainen, M. (1995) "Polycontextuality and boundary crossing in expert cognition: Learning and problem solving in complex work activities," *Learning and Instruction* (5:4), pp. 319-336.

Faraj, S. and Sproull, L. (2000) "Coordinating Expertise in Software Development Teams," *Management Science* (46:12), pp. 1554-1568.

Fischer, G. and Ostwald, J. (2005) "Knowledge Communication In Design Communities," in: *Barriers and Biases in Computer-Mediated Knowledge Communication*, R. Bromme, F. Hesse, and H. Spada (eds.). New York, NY: Springer, pp. 213 - 242.

Flor, N.V. and Hutchins, E.L. (1991) "Analyzing distributed cognition in software teams: a case study of team programming during perfective software maintenance," *Empirical Studies of Programmers - Fourth Workshop Norwood NJ: Ablex*, J. Koenemann-Belliveau, T. Moher, and S. Robertson (eds.), pp. 36-59.

Gasson, S. (2006) "A Genealogical Study Of Boundary-Spanning IS Design," *European Journal of Information Systems* (15:1), pp. 26-41.

Gasson, S. (2007) "Progress And Breakdowns In Early Requirements Definition For Boundary-Spanning Information Systems," *ICIS '07*, Montréal, Québec, Canada.

Gero, J.S., Tham, K.W., and Lee, H.S. (1992) "Behaviour: A Link Between Function and Structure in Design," in: *Intelligent Computer Aided Design*, B. DC, W. MB, and Y. H (eds.). Amsterdam, NL: Elsevier, pp. 193-225.

Gersick, C.J.G. (1988) "Time And Transition In Work Teams: Toward A New Model Of Group Development," *Academy of Management Journal* (31:1), pp. 9-41.

Gersick, C.J.G. (1991) "Revolutionary Change Theories: A Multilevel Exploration of the Punctuated Equilibrium Paradigm,," *Academy of Management Review* (16:1), Jan., 1991, pp. 10-36.

Greenbaum, J. and Kyng, M. (1992) "Introduction: situated design," in: *Design at work: cooperative design of computer systems*, J. Greenbaum and M. Kyng (eds.). pp. 1-24.

Guindon, R. (1990) "Designing the design process: Exploiting opportunistic thoughts," *Human-Computer Interaction* (5:2/3), pp. 305-344.

Heidegger, M. (1962) *Being and Time*. New York NY.: Harper & Row New York.

Hirschheim, R.A. (1986) "Participative Systems Design User Experiences Evaluation And Conclusions," *The Australian Computer Journal* (18:4).

Hollan, J., Hutchins, E., and Kirsh, D. (2002) "Distributed cognition: toward a new foundation for human-computer interaction research," *ACM Transactions on Computer-Human Interaction (TOCHI)* (7:2), pp. 174 - 196.

Hutchins, E. (1995) *Cognition in the Wild*. Bradford: MIT Press.

Land, F. (1992) "The Information Systems Domain," in: *Information Systems Research*, R. Galliers (ed.). Oxford UK: Blackwell Scientific, pp. 6-13.

Lave, J. (1988) *Cognition In Practice: Mind Mathematics And Culture In Everyday Life*. Cambridge UK: Cambridge University Press.

Lave, J. and Wenger, E. (1991) *Situated Learning: Legitimate Peripheral Participation*. Cambridge UK: Cambridge University Press.

Lawson, B. (2000) *How Designers Think: The Design Process Demystified*, (3rd. ed.). Oxford UK: The Architectural Press.

Levine, J.M. and Moreland, R.L. (1999) "Knowledge Transmission in Work Groups: Helping Newcomers to Succeed," in: *Shared Cognition In Organizations: The Management of Knowledge*, L. Thompson, J.M. Levine, and D.M. Messick (eds.). Mahwah, NJ: Lawrence Erlbaum Associates, pp. 267-296.

Lin, A. and Silva, L. (2005) "The social and political construction of technological frames," *European Journal of Information Systems* (14), pp. 49-59.

Liu, K., Sun, L., and Bennett, K. (2002) "Co-Design of Business and IT Systems—Introduction by Guest Editors," *J. of Information Systems Frontiers* (4:3), pp. 251-256.

Maher, M.L. and Poon, J. (1996) "Modelling design exploration as co-evolution," *Microcomputers in Civil Engineering* (11:3), pp. 195-210.

Maher, M.L. and Tang, H.-H. (2003) "Co-evolution as a computational and cognitive model of design " *Research in Engineering Design* (14:1), pp. 47-64.

Malhotra, A., Thomas, J., Carroll, J., and Miller, L. (1980) "Cognitive Processes In Design," *International Journal of Man-Machine Studies* (12), pp. 119-140.

Markus, M.L., Majchrzak, A., and Gasser, L. (2002) "A Design Theory For Systems That Support Emergent Knowledge Processes," *MIS Quarterly* (26:3), pp. 179-212.

Mintzberg, H. and Waters, J.H. (1985) "Of Strategies Deliberate and Emergent," *Strategic Management Journal* (6:3), pp. 257-272.

Mumford, E. (2003) *Redesigning Human Systems*. Hershey, PA: Idea Group.

Nardi, B. (1995) "Studying context: a comparison of activity theory, situated action models, and distributed cognition," in: *Context and consciousness: activity theory and human-computer interaction* B. Nardi (ed.). Cambridge MA: MIT Press, pp. 69-102.

Newman, M. and Robey, D. (1992) "A Social Process Model of User-Analyst Relationships," *MIS Quarterly* (16:2), pp. 249-266.

Orlikowski, W.J. and Gash, D.C. (1994) "Technological Frames: Making Sense of Information Technology in Organizations," *ACM Transactions on Information Systems* (12:2), April 1994., pp. 174-207.

Pries-Heje, J. and Baskerville, R. (2006) "Designing Organizational Change in IT: A Theory Nexus," *Design Science Research in Information Systems and Technology (DESRIST 2006)*, S. Chatterjee and A.R. Hevner (eds.), Claremont CA: Claremont Graduate School [http://ncl.cgu.edu/designconference/].

Purao, S., Truex, D.P., and Cao, L. (2003) "Now The Twain Shall Meet: Combining Social Sciences And Software Engineering To Support Development Of Emergent Systems," *Americas Conference on Information Systems (AMCIS)*, Tampa FL.

Rittel, H.W.J. (1972) "Second Generation Design Methods," in: *Developments in Design Methodology* N. Cross (ed.). Chichester UK: J. Wiley & Sons (1984), pp. 317-327.

Rittel, H.W.J. and Webber, M.M. (1973) "Dilemmas in a General Theory of Planning," *Policy Sciences* (4), pp. 155-169.

Scarbrough, H. and Corbett, J.M. (1991) *Technology and Organisation: Power Meaning and Design*. UK: Routledge.

Simon, H.A. (1973) "The Structure of Ill-Structured Problems," *Artificial Intelligence* (4), pp. 145-180.

Simon, H.A. (1996) *The Sciences of The Artificial*, (3rd. ed.). Cambridge MA: MIT Press

Star, S.L. (1989) "The Structure of Ill-Structured Solutions: Boundary Objects and Heterogeneous Distributed Problem Solving," in: *Distributed Artificial Intelligence, Vol. II.*, L. Gasser and M.N. Huhns (eds.). San Mateo CA: Morgan Kaufmann, pp. 37-54.

Suchman, L. (1987) *Plans And Situated Action*. Cambridge MA: Cambridge University Press.

Truex, D.P., Baskerville, R., and Klein, H.K. (1999) "Growing systems in emergent organizations " *Communications of the ACM* (42:8), pp. 117-123.

Truex, D.P., Baskerville, R., and Travis, J. (2000) "Amethodical Systems Development: The Deferred Meaning of Systems Development Methods," *Accounting Management and Information Technologies* (10:1), pp. 53-79.

Turner, J.A. (1987) "Understanding The Elements Of Systems Design," in: *Critical Issues In Information Systems Research*, R.J. Boland and R.A. Hirschheim (eds.). New York, NY: Wiley, pp. 97-111.

Urquhart, C. (2001) "Analysts and clients in organisational contexts: a conversational perspective," *The Journal of Strategic Information Systems 10* (3:243-262.).

Valkenburg, R. and Dorst, K. (1998) "The reflective practice of design teams," *Design Studies* (19:3), pp. 249-271.

Weick, K.E. (1987) "Organizational Culture As A Source Of High Reliability," *California Management Review* (29:2), pp. 112-127.

Weick, K.E. (1995) *Sensemaking In Organizations*. Thousand Oaks CA: Sage.

Winograd, T. and Flores, F. (1986) *Understanding Computers And Cognition*. Norwood New Jersey: Ablex Corporation.

Endnotes

1. The concepts of *problem-space* and *solution-space* derive from the psychology of programming and design literature. Initially, the problem-space concept was proposed as part of an algebraic representation of external problem-structures that existed independently of the analyst (Newell & Simon, 1972). More recently, the concepts have been used more fluidly, to denote a mental model, or conceptual "space" of potential design problems or solutions that underlie the selection of specific design attributes (Dorst & Cross, 2001). The concept is developed further by Bergman et al (2002a & 2002b), as discussed in the paper. to distinguish between the solution-space as it exists now (i.e. the ecology of functional requirements that solves current perceptions of the problem-structure) The fitting of solution spaces to problem spaces as indicative of cognitive frames in design is also found in the model underlying Alexander's (1979) pattern language, that has more recently been applied to object domain theory. Agent theory and HCI design (Alexander, 1999).