A CONCEPTUAL FRAMEWORK FOR INFORMATION SYSTEM DEVELOPMENT
– A DECISION-MAKING PROCESS PERSPECTIVE

Chris Barry

Department of Accountancy and Finance
National University of Ireland, Galway.

University Road
Galway
Direct Dial: + 353 - 91 - 750301
email: chris.barry@nuigalway.ie

1. Introduction

Academics and practitioners have been increasingly aware that there is a credibility gap between claims made for the benefits and usage of information systems (IS) methodologies and the actuality of systems development practice. Apparent breakthroughs in development practices such as structured methods and later object-oriented methods have done little to simplify the process. There have been hundreds of methodologies (mainly commercial) proposed over the past three decades, often used with limited success. The prospect or expectation that new methodologies would ease the burden of development has never really been realised - many aspects of it are as chaotic and unstructured as ever. Far from being simplified, systems development has become ever more complex, as systems functionality grows to meet organizational and user needs. Furthermore, firms are buying in rather than building software wherever possible – significantly changing the component parts of IS, elaborating activities such as systems integration, project management and outsourcing management.

2. Differing Schools on ISD

While there are many perspectives on how IS should be developed, three important school of thought are discussed below. Although the general “hard” or “soft” debate has been ongoing for some time now, it is a discourse that represents fundamental philosophical differences that are significant for both academics and practitioners. The third and most recent school of thought is the “amethodical” school, one that like the “soft” approach to systems development, places little store in formal methods.

2.1. The Normative View of Systems Development and Methodologies

The normative view that quality, productivity and user needs would all be improved by the use of structured methodologies has been a dominant one. On the face of it, it is not unreasonable to suggest that adding structure to a complex process is a rational means of managing the activities of information systems development (ISD). Indeed there are many tasks such as project management, systems analysis and design techniques and
personnel management that remain prerequisite activities during the development process. The perspective has largely grown from a rationalist view of systems, and thus IS, as being objective, understandable artefacts. This positivist approach to systems development remains the overt method used by most organisations. It is considered logical and sensible for the actors involved in a project (managers, users, systems analysts, programmers, etc.) to expect an orderly process that delivers a system on time, within budget and meets the needs of those commissioning it. This mechanistic approach has its roots in scientific problem solving and early management theories expressed by the likes of Taylor (1947) and embedded in early and indeed later structured systems analysis texts widely used by practitioners (DeMarco, 1979; Yourdon, 1989). In truth, most systems are (overtly) developed like this - but encounter difficulties that this author believes are inherent in the formal, structured, methodological approach widely used in systems development.

While the structured methods remain popular, there is growing evidence that practitioners are not using them imitatively or slavishly. In an exploratory study Bansler and Bodker (1993) looked at how closely developers worked with the prescribed methodology, Yourdon and DeMarco’s “Structured Analysis” (DeMarco, 1979). They found that while certain techniques are used (primarily data flow diagrams), developers simply do not follow the procedures prescribed by the method. Their findings add weight to the contention that the normative view of systems development in technical and much academic literature is inconsistent with the way in which it is carried out in practice. Other early work has suggested similar, contrary views to a normative outlook on ISD. Keen’s suggested ISD was as much a political process as well as a technical one (Keen, 1981).

In a recent study of development methods, it was revealed that while developers had no difficulty in using or understanding methodologies for developing conventional (rather than multimedia or Web-based) systems, it was reported by 61.5% of respondents that they were simply very cumbersome to use (Barry and Lang, 2001). Many also felt that commercial methodologies were too costly and not very well suited to the real world. With multimedia and Web-based systems development the additional complexity of designing IS that often have an external focus and complex back-end integration with existing systems, the problem is even more difficult. For these types of systems what methods should developers use? There have been several methods put forward that have dominated the academic literature (Garzotto et al., 1993; Isakowitz et al., 1995; Gellersen et al., 1997). However research indicates that for the most part developers are not using them (Barry and Lang, 2001). There is also evidence of method tailoring — rational behaviour that leads developers to choose parts of different methods depending on the contingent circumstances of the development context (Fitzgerald et al., 2000).

Until recently there has been a working assumption that without structured methods developers are using poorly disciplined approaches to ISD, particularly Web-based IS development. Lang and Fitzgerald (2003) have challenged this unsafe assumption. Questions about a looming hypermedia crisis have not being borne out by their research that reveals that hypermedia and Web-based development is much more disciplined than is commonly believed. Understanding practice rather than proposing yet more prescriptive models should be a key starting point if the development community is to be well served by academic research.
Despite the aforementioned problems with structured methods there is increasing pressure on organisations to formalise the ISD process (Fitzgerald et al., 2002). For example the attractiveness of ISO certification, the required use of standardized methods by governments in certain countries and the Capability Maturity Model (CMM) used in software engineering.

2.2. Alternative “Soft” ISD Methods

Alternative views have co-existed for some time now, from soft systems methodology (SSM) by Checkland (1981), information systems work and analysis of change (ISAC) by Lundeburg (1982), ETHICS by Mumford (1983) to more recent revisions such as Multiview 2 (Avison et al., 1998). These methods are considered “soft” in contrast to the “hard” systems development approaches or traditional operations research (OR) techniques. They share a certain philosophical position that IS are socio-technical, made up of people, machines and processes and that the human dimension is not adequately addressed with structured methods. Collectively these methods have been called problem structuring methods – PSMs (Rosenhead and Mingers, 2001), a useful expression since this activity is largely what they address. They deal variably with issues such as arbitration, reducing complexity, improving understanding and the resolution of social conflict (not just in ISD but in any problem solving context). Systems are considered subjective and are therefore interpreted differently depending on the participant’s agenda or bias (thus the “interpretivist’s” standpoint). In ISD this is both their strength and weakness. Many of the soft approaches are theoretically sound but difficult to implement in the real world. While it is possible for proponents of these methods to point to actual implementations, such applications are typically well-documented case studies within the literature of the “soft systems school”. In reality for every system developed using PSMs there are thousands developed with conventional methods. Perhaps related to the last point, the usability and relevance of the methods is questionable. Although they have been around for many years now, few organisations actually use them. There may well be a widening gulf between new theoretical constructs generated by academics and the ability or time that practitioners have to understand or implement them.

The second major problem with the use of PSMs for ISD is that they primarily deal with the systems analysis stage of the systems development life cycle (SDLC) unlike most of the structured or object-oriented methods. In practice sooner or later a set of agreed requirements must be fully articulated as a physical design specification and then programmed, assembled with other components and implemented. The fit between PSMs and design specification techniques and development tools is poor. Few of these methods try to fuse the PSM with the remainder of the SDLC.

2.3. The Amethodical School

Software development has been termed a “wicked problem” (Rittel, 1972) as opposed to a tame one. It is complex, unpredictable and there is no clear procedure for resolving all the problems that arise during systems development. A growing body of opinion now sees the ISD process in defiance of method, beyond the assistance of conventional approaches and essentially amethodical (Suchman, 1987; Baskerville et al., 1992; Introna and Whitley, 1997). From the outside the design activity may appear as chaotic
and perhaps slightly out of control but the process is directed by the “hidden rationality of skilled individuals” (Conboy et al., 2002). Anarchy is avoided through experienced decision-making, or “smart improvisation” (Ciborra, 1999). Ciborra as describes it thus:

“Improvisation is simultaneously rational and unpredictable; planned but emergent; purposeful but opaque; effective but irreflexive; discernible after the fact, but spontaneous in its manifestation”.

What is being described is not the absolute absence of order but reflective activity that contributes to the improved effectiveness of a project, rather than improvising simply to dispense with formalised methods. Implicit in this perspective is that individuals are not seeking to develop a perfect system and that there is some sub-optimal decision-making taking place because for example there is incomplete information, limited time or there are individuals constraints of “bounded rationality” (Simon, 1981). Developers are achieving satisfactory solutions rather than “the best”. In the fog that usually accompanies ISD projects, there will always be circumstances where Lindblom’s “science of muddling through” (Lindblom, 1959) is used to reach an acceptable outcome.

While the use of an improvisational approach to ISD may be more intuitively attractive than a scientific, rationalist one there are problems in relying entirely on the serendipitous consequences of the absence of method. In work by Conboy et al. (2002) a number of problems are identified as inherent in the approach. Important knowledge learnt by developers is more difficult to pass on to less experienced colleagues. If the reasoning behind particular good or bad decisions is not captured, how can apprentices be efficiently educated? Improvisation also causes problems with the management of project. Improvisation assumes a greater degree of independence for developers and ipso facto less control by managers. If control is shifted towards the developers, how is a project’s progress assessed, what are the key issues hindering the development and how is quality being measured? A third issue is that improvisation can actually encourage developers to “embrace their biases to the point that alternative views are occluded” or to “inflate the importance of their own point of view at the expense of others” (McPhee, 1997).

3. Another Perspective on ISD

Despite extensive research on the nature of ISD, this paper takes note of the failure of new ISD methodologies and theories to reach wide agreement amongst practitioners or within academic circles and of the shortcoming in each of the three approaches discussed above. A different perspective on ISD is considered here, one that draws on long-standing but perhaps overlooked IS research. Management decision-making was central to intense academic activity in decision support systems (DSS) during the 1970s and 1980s and executive information systems (EIS) during the 1980s and 1990s. Insights from decision-making theory led to frameworks and widely agreed perspectives on the nature of such systems and how they should be developed (Gorry and Scott Morton, 1971; Sprague, 1980; Rockart and Treacy, 1982; Watson et al., 1991). The importance of decision-making theory in defining the activities in the DSS development process has not been generalised to yield insights into more conventional, larger-scale systems or indeed Web-based IS (Barry, 2000). In the same way that decision-making is a central part of management activity, ISD is continually confronted
and moved forward by decision-making actions on the part of the participants or actors in the development process. This happens whether the decision maker is guided by method or inspired by improvisation.

Thus a critical context for ISD is the decision-making “posture” of all those involved in a systems development project. If this is assumed to be so then the classical models of decision-making identified by Keen and Scott Morton (Keen and Scott Morton, 1978) are essential to our understanding of the IS development process in the same way that they added greatly to our understanding of DSS design. When decision-making, organisational theory and the roles and motivations of various actors are considered a more complex and ambiguous ISD perspective emerges.

3.1. ISD as a Decision-making Process

ISD is a process, conventionally viewed as the SDLC, and within it are various tasks and activities that must be carried out to bring a project to a successful conclusion. There are as many versions of it as there are commercial or in-house methodologies. For the purposes of simplicity the four essential stages - systems analysis, systems design, construction and implementation are used here. A simple version of the decision-making process is also used to illustrate the correspondences between the two “processes”. Figure 1 below draws parallels between them, demonstrating obvious similarity. Systems analysis and implementation corresponds directly with the notion of intelligence gathering leading to an established set of requirements and the final implementation of the system or decision. Systems design and construction would roughly equate to the design and choice stages of the decision-making process. The key conclusion to be drawn from this analysis is that, at a macro level, the SDLC can be viewed as a process that closely resembles decision-making. Even alternative approaches such as evolutionary development or a middle-out approach can be shown to have similar correspondences. Each iteration of the system would for example include all the stages of the decision-making process.

<table>
<thead>
<tr>
<th>Systems Development Life Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stages</td>
</tr>
<tr>
<td>“Classical” Decision Making Process</td>
</tr>
<tr>
<td>Intelligence</td>
</tr>
<tr>
<td>On-going Decisions</td>
</tr>
</tbody>
</table>

Figure 1: Correspondence Between the Decision-making Process and the SDLC

The figure above also illustrates there are natural, on-going decisions that are faced by actors working on the project, independently or collectively, by negotiation or by fiat.
3.2. Models of Decision-Making

In a seminal work on decision support systems, Keen and Scott Morton identified five models of decision-making (Keen and Scott Morton, 1978). The models or views range from the wholly normative to entirely descriptive. The approaches are not mutually exclusive and some will be relevant for certain participants in the ISD process and others will not. Each view points to important issues that should be understood by the development community, particularly management. While it has long been the case (and for many it still is) that IS developers were committed to the normative, rationalist paradigm some changes have been evident in more recent times. The models are briefly summarized below.

1. The Rational View
   - The classical notion of objective decision-making with complete information, based on early microeconomic theory. It is a normative view that dominated the development of IS and operations research/management science (OR/MS) for decades.

2. The “Satisficing” Process-Oriented View
   - This view holds that in the real world we cannot know all possible outcomes and are therefore unable to choose an optimal one. Thus we are left with choosing a satisfactory one based on judgement or heuristics. It also recognises constraints such as time and costs.

3. The Organisational Process View
   - Based on the formal and informal organisation, how it operates and what its lines of communications are. The view is “an extension of the intendedly rational approach” (Keen and Scott Morton, 1978) while accepting there will be bargaining between competing organisational units.

4. The Political Process View
   - A pluralist perspective that recognises the natural diversity of objectives, interests and views in any organisation and their influence on the decision-making process. It is the antithesis of the rational approach and expressly articulates what many people intuitively sense about decision-making.

5. The Individual Differences Perspective
   - Focuses on the individual as being unique with distinct abilities and decision-making styles. Aspects of this view are that individuals have varying levels of cognitive skills in problem solving and what is useful information to one person may be of no use to another. For the purposes of ISD the view informs us that people have different cognitive styles and approaches to problem solving.

The models outlined above are the starting point for the framework developed here. Other models of decision making could have been chosen, such as Bahl and Hunt’s decision-making descriptive models (Bahl and Hunt, 1984), however the resilient models of Keen and Scott Morton justify their use within the framework.
3.3. **Actors in the ISD Process**

The traditional composition of those involved in an ISD project is:

- Management/Steering Committee
- Users
- IS Managers
- Programmers/Software Engineers
- Systems Analysts/Designers
- Network specialists

In more recent times the individuals and groups involved in a systems development project has become more extensive. This is because of due recognition of the integral role of users, those affected by the system and more diverse development teams deployed in multimedia and Web-based IS. For such projects there are roles for:

- Managers
- Animators/Graphic Designers
- Programmers/Software Engineers
- Systems Analysts/Designers
- Scriptwriters/Storyboarders
- Users
- Audio Producers
- Video Producers
- Localization/Technical Writers
- Technical specialists

Research indicates that Animation/Graphic Design, Programming/Software Engineering, and Systems Analysis & Design are major roles to which professionally trained individuals are normally singly dedicated (Barry and Lang, 2003). There are lesser roles for others, where there will be some overlap - for example, the same person might perform Audio and Video Production. Ironically, there may be a significantly reduced role for users if the project is a commercial one where the user base is not close at hand. In such cases the commissioner of the project typically fills the role of user or “specifier” of requirements. However for the purpose of work herein only in-house IS projects are the subject of this analysis. Real-world project might be further complicated as the more widespread outsourcing of IS projects brings contractors and their teams into the frame.

4. **The Conceptual Framework**

The conceptual framework put forward combines the analysis of explicit decision-making within the SDLC, the key models of decision-making and the actors involved in the development process. It is illustrated in Figure 2 below. The three aspects of development are combined to reveal a three-dimensional model charted on three axis X, Y and Z. The axes represent:

- X axis – Decision-making models
- Y axis - Decision-making stages
- Z axis - Actors

To simplify the framework (and make it more visually meaningful) only certain individuals or groups have been included although as pointed out in section 3.3 above the actors involved in projects are increasingly diverse. Furthermore, they may be
several actors within a particular category, say analysts identified as \{analyst_1, analyst_2, … analyst_n\}. For every intersection of the three axes within the framework it is possible to identify the posture of the individual or group in a given phase of the project and within the context of a decision-making style.

Figure 2: The Conceptual Framework

These postures yield a rich picture of how individuals or groups may hold differing perspectives depending on the decision-making model they subscribe to. What the framework immediately demonstrates is that there will be co-existing decision-making paradigms amongst actors during an ISD project. This simultaneity of perspectives exposes postures that may agree or disagree. A profile of all actors on the project can draw precise notice to tensions that exist between actors, suggesting remedial action or the general outlook that might dominate an organisation’s posture toward systems development. Examples of postures held by actors are shown in Figure 3. For each the actor, their decision-making approach and the posture they hold in each stage of the decision-making stages and aspects of the development process are illustrated. These aspects of the development process correspond to the simplified decision making stages outlined earlier. For each stage the posture of the actor is illustrated regarding; what their objective and expectations are for the system; how they view the means by which it is developed; and what end or output they expect from the system. For example the IS Manager acting rationally might establish “objective” organisational and user needs and be very firm on the systems budget; the means by which the systems is developed is important, and process and project management critical, while he or she views the output as a software product, an IS system or a service delivered within budget and on time. This IS Manager would not be content with an imperfect system.
<table>
<thead>
<tr>
<th>Actor: IS Manager</th>
<th>DM View: Rational View</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DM Process</strong></td>
<td>Intelligence</td>
</tr>
<tr>
<td>Development</td>
<td>Objective</td>
</tr>
<tr>
<td>process</td>
<td></td>
</tr>
<tr>
<td><strong>Posture</strong></td>
<td>Establish objective organisational and user needs, budget-driven</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actor: IS Manager</th>
<th>DM View: Satisficing View</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DM Process</strong></td>
<td>Intelligence</td>
</tr>
<tr>
<td>Development</td>
<td>Objective</td>
</tr>
<tr>
<td>process</td>
<td></td>
</tr>
<tr>
<td><strong>Posture</strong></td>
<td>Establish acceptable organisational and user needs, budget-flexible depending on acceptance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actor: Analyst</th>
<th>DM View: Individual Differences View</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DM Process</strong></td>
<td>Intelligence</td>
</tr>
<tr>
<td>Development</td>
<td>Objective</td>
</tr>
<tr>
<td>process</td>
<td></td>
</tr>
<tr>
<td><strong>Posture</strong></td>
<td>Establish subjective user needs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actor: The Formal Organisation</th>
<th>DM View: The Organisational Procedures View</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DM Process</strong></td>
<td>Intelligence</td>
</tr>
<tr>
<td>Development</td>
<td>Objective</td>
</tr>
<tr>
<td>process</td>
<td></td>
</tr>
<tr>
<td><strong>Posture</strong></td>
<td>Establish systems objectives consistent with organisational objectives</td>
</tr>
</tbody>
</table>

**Figure 3: Illustrated Postures of Actors**
The model can be further elaborated to illustrate two phenomena, essentially atypical postures that are difficult to represent or conceptualise with any of the three general approaches discussed earlier. Firstly, during the development process the decision-making models used by various actors may undergo change. For example, at an early stage of development analysts may adopt (formally at least) a rational decision-making stance while later a satisficing one becomes more realistic. This phenomenon is illustrated in Figure 4 (a). It represents postures taken consecutively \(\{X_1,Y_2,Z_3\} + \{X_2,Y_2,Z_5\}\) by the actor.

The following are posited as the most likely decision-making approaches actors might use:

- Management – rational, satisficing, organisational process, political.
- Users – organisational process, political, individual differences.
- The Formal Organisation - organisational process.
- IS Managers - satisficing, political, individual differences.
- Analysts - rational, satisficing, political, individual differences.
- Programmers – rational, satisficing, individual differences.

The second phenomena (illustrated in Figure 4 (b)) the framework can reveal is that an actor may hold different decision-making styles simultaneously. This may happen for a number of reasons. A programmer, during design, who might outwardly subscribe to the software engineering line on rationalism and structured development methods may in fact as Parnas and Clements put it (1986) be “faking the ideal process”. The programmer may be taking short cuts, not documenting the system or even “borrowing” code from somewhere else. For whatever reason this actor is caught between overt and covert decision-making styles. The simultaneously held postures are denoted as \(\{X_1,Y_2,Z_6\} + \{X_2,Y_2,Z_6\}\). In a somewhat similar way a systems analyst may hold a firm set of beliefs about how ISD should be carried out, let’s say a wholly political one, but act in a completely different manner because for example the commissioning organisation has decreed that SSADM must be used for the systems development project. Thus actors during development might have a range of perspectives that overlap, are explicit or perhaps hidden.
The relationship between the various postures held by the actors on the development “stage” are complex – the formal organisational posture may be deterministic or normative based upon a rationalist paradigm, embraced by management at one level but who are also wholly aware of the political exigencies or necessary, media-related obfuscation that digress from an idealised vision. In practice this might mean that an organisation believes strongly in the overt use of “method-ism” (Whitley, 1998) while using social, political, arbitration and improvisational techniques to resolve problem situations. While this should not greatly surprise anyone, it is difficult to express these conceptually. It is intended that the framework may be a starting place for demonstrating both the pragmatic incrementalism and the plurality of organisational decision-making in developing IS.

As the systems development process advances, the design elements change from abstract constructs to concrete artefacts. The degree of concreteness during the SDLC is illustrated in Figure 5. The shift in some actors’ decision-making behaviour may mimic the way a system development process is generally abstract at an early stage because user requirements are unclear or unknown but eventually becomes concrete or hard-coded. It also explains why analysts might be more political or satisficing in their behaviour because much of what they do, particularly early in a project, requires social interaction, conflict resolution skills and political astuteness. On the other hand programmers and technical specialists are more likely to be rationalist or take the organisational process view in their problem solving behaviour.

<table>
<thead>
<tr>
<th>Systems Development Life Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stages</td>
</tr>
<tr>
<td>Degree of Concreteness</td>
</tr>
</tbody>
</table>

**Figure 5: Degree of Concreteness during the SDLC**

5. Conclusions

The main contention of the framework is that in essence, IS development is a decision-making process and that a deep understanding of differing viewpoints on systems development held by actors can be demonstrated by it. The framework clearly illustrates that systems development may be in part deterministic, in part without structure - it is creative yet somewhat ordered, improvisational yet explicit, and both rational and political. Indeed, incremental decision-making may explain much of the reality of systems development. The framework also suggests that competing ISD theories, positivism and interpretivism, co-exist within a project and within actors working on it. The framework thus is a tentative, bringing together of the three approaches or schools of thought on ISD outlined above in sections 2.1 to 2.3. The debate, often heated, has at times clouded the reality of just how complex, and
sometimes contradictory, IS development really is. The framework demonstrates the pluralism of ISD - the normative view explicit in the conventional methodological approach contrasted with the descriptive reality that underlies the PSMs. Similarly, while formalised methodologies can be overly visible the opposite is the case with improvisational decision-making but they may still co-exist.

There are limitations with the framework – it is conceptual and by its nature is not going to be a tool or method of analysis for practitioners. It also needs refinement to more fluently illustrate the co-existence of differing ISD schools of thought. Finally, it does not yet suggest how best IS should be developed. Despite these shortcomings the framework can yield, through the analysis of postures, rich visual representations of complex interactions, between actors, their decision-making approach and the ISD process itself.

References


