Design Researcher's IS Artifact: a Representational Framework

Rustam Vahidov

Dept. of Decision Sciences & MIS
John Molson School of Business
Concordia University
Montreal, Quebec H3G 1M8
Canada

E-mail: rvahidov@jmsb.concordia.ca

Abstract

Design research has gained much interest among the academicians of Information Systems. Unlike research in natural and social sciences, design research does not have a long established history and tradition. The methodological foundations for this type of research are emerging through contributions in various academic IS outlets. The argument underlying this work is that the attempts to define the methodology for the design science have to be complemented (perhaps, even preceded) by the advancement of the adequate representational framework. In other words, in addition to or before asking "how" question, a proper consideration needs to be given to "what" question. This work aims at introducing such a framework for structuring representation of a design researcher's artifact.

Keywords: Design Research, Representational Framework, Zachman's Framework, Ontology of IS, System Development

1. Introduction

"Design science" or "design research" is becoming the powerful trend in the study of Information Systems (Au 2001; Ball 2001; Burstein and Gregor 1999; Hevner et al. 2004; March and Smith 1995; Markus et al. 2002; Walls et al. 1992). Unlike "natural science" type of research (March and Smith 1995), design researchers aim at producing solutions that can be employed in various organizational contexts to achieve specific objectives. In other words this type of research is largely utility-centric, as opposed to being explanation-centric.

At the present time the scientific status of and methodology for design research are being established, for example by dignified efforts in clarifying the outputs of such research (March and Smith 1995), the theories of design (Walls et al. 1992), and guidelines for the academicians conducting design-centric projects (Hevner et al. 2004).

DESRIST 2006. February 24-25, 2006, Claremont, CA.

One issue that necessarily arises when inquiring into the nature of design research is "what makes particular design projects qualify as a research?" In other words, how could one show, that a given artifact represents knowledge rather than a particular solution. Since a distinguishing characteristic of knowledge is its generality, the necessary condition for design to be considered research is also the generality of the outcomes produced. Thus, in design *research* the focus should never be on particular instantiations (though these too, are of importance as illustrations/confirmations), but on generic systems, abstract systems, or classes of IS artifacts. This is implicit in the formulation of design theories, with their major components including *type* of requirements, and *type* of system solutions (Walls et al. 1992), as well as in the notion of "*meta*-artifacts" (Livari 2003).

If we accept that the significant output of design research is a generic IS artifact, the next important question is how to represent such systems? The frameworks, techniques and tools for representing concrete, specific information systems are well-established and are being employed intensively by the design practitioners. However, the question of finding an appropriate framework for representing classes of IS to be employed by design researchers remains largely unexplored.

The objective of this paper is to propose a framework for representing wide classes of information systems for the design researchers in IS. Thus, the work seeks to explore the ontological, rather than methodological aspects of design research. In doing so, we will adopt Zachman's framework for representing (concrete) information systems as a starting point and subsequently modify it to fit the researchers' needs. We will further illustrate the application of the framework to represent a new design concept for decision support borrowed from the literature.

2. Motivation and Background

"We should help create the future, not just study the past"
Paul Gray, in (Kock et al. 2002)

Herbert Simon's seminal book "The Sciences of the Artificial" has raised the issue of design-type of research in the context of computing disciplines (Simon 1996). One of the powerful themes in the book is to promote the acceptance of the sciences of the artifacts. Simon had introduced the concept of an artifact through the separation of its outer environment from the inner environment. The outer environment could be regarded as the requirements imposed on the artifacts' function. The inner environment is the internal organization of the artifact. Simon regarded design as the interface between the outer and inner environments of an artifact. This could be regarded as the most general representational schema for an artifact.

March and Smith in their influential paper contended that both descriptive and prescriptive perspectives in IS research can be accommodated under the broad notion of Science (March and Smith 1995). In particular, they showed how these two kinds of research could interact and mutually enrich each other. As for design research, they have identified four types of products, including: constructs (language), models, methods, and implementations (the latter included both prototypes as well as working systems).

Walls et al. have conceived the concept of a "design theory" for information systems (Walls et al. 1992). They envisaged the use of "kernel theories" in the design of a *class* of artifacts. According to them design theories should consist of type of user requirements ("meta-requirements"); type of system solution or class of artifacts ("meta-design"); and the type of methodology used to develop such artifacts. They had used an example of "vigilant EIS" to illustrate a design theory. More recently, Markus et al. employed this concept of a design theory to devise the characteristics of a broad class of systems that the authors called "systems that support emergent knowledge processes" (Markus et al. 2002).

Hevner et al. have noted that IS researchers are uniquely positioned to perform design type of research that relies both on organizational science and technology (Hevner et al. 2004). They stressed the necessity of emphasizing rigor in conducting design research and advocated a number of guidelines for design researchers. One of their major emphases was on the originality of such works. Albert et al. have illustrated the application of these guidelines in their work on the design of customer-centric websites (Albert et al. 2004).

The above contributions as well as some others in this direction are truly exciting to design researchers who seek to establish the rigorous scientific platform to base their endeavors upon, and adopt a solid scientific methodology in conducting design-centric projects. However, for the IS researchers one crucial question remains open: that of representing the artifacts. Instances of information systems are complex artifacts with numbers of perspectives/stakeholders and various aspects of the system forming a multitude of facets, each descriptive of the IS and relevant in its own right. The practitioners of various types rely heavily on a number of representational tools, constructs, frameworks in different phases of IS development and maintenance. Could a representational framework be useful to IS design researchers, whose focus is on classes of systems, as well? In our opinion, the answer is a definitive "yes".

There are at least two good reasons to justify the necessity for a representational framework for the classes of IS (meta-artifacts). First, the different major aspects of the system (i.e. structural, dynamic) must be brought out to address the completeness of the description of the design researcher's artifact. In other words, various characteristics, or descriptive facets of the system must be identified. Second, the appropriate level of refinement of the artifact in terms of technical details should be brought in to help manage the scope of research designer's projects. In this regard, instantiated systems are not or should not be the primary focus for the researchers. The design researcher's output, as that of any other researchers, should be knowledge, and concrete systems cannot qualify as knowledge, though they are very much relevant as *observations*. Then the question arises as to what level the technical detail should be addressed in one's research projects. The remainder of this paper will propose one such representational framework for abstract IS artifacts.

3. Zachman's Framework for Information Systems Architecture

Since, to our knowledge little work has been done for representing design researchers' artifacts, we will start building our model based on the frameworks for describing concrete information systems. Practitioners of IS have long recognized the value of representational frameworks, in particular for communicating the vision and enabling the evolution of enterprise systems (Armour et al. 1999; Sowa and Zachman 1992; Zachman 1987). To this end a number of frameworks had been advanced in the past.

One particularly interesting framework that seems to suit our needs had been proposed by Zachman, and subsequently extended by Sowa and Zachman (Sowa and Zachman 1992; Zachman 1987). Zachman had recognized the level of complexity inherent in enterprise information systems and devised an elegant framework which could represent systems in an organized fashion. His framework can serve as a prism through which various stakeholders can gain a relatively complete picture of a particular IS according to their level of description. The two-dimensional matrix organization of this model is the major motivation for us to adopt it as a starting point, as it neatly addresses the points stressed earlier.

One dimension of the Zachman's framework (rows) captures perspectives of various stakeholders, including Planner (scope), Owner (enterprise model), Designer (system model), Builder (technology model), Subcontractor (out-of-context model), and, finally, functioning system. Zachman's insight was that any given IS can be *completely described* from each of the above perspectives. The second dimension (columns) then, introduces the pertinent categories that can be used for such descriptions. These are conceptualized as "different abstractions from or different ways to describe the real world" to contain the complexity of information systems (Sowa and Zachman 1992).

The categories include Data (what), Process (how), Network (where), People (how), Time (when), and Motivation (why). The Data category describes the system as entities along with their structural relationships for a given level of representation. The Process category models the dynamic properties of the system. The Network column captures the fashion in which system architecture is distributed. The People column describes the human and organizational contexts of an IS (organization chart, human interface, security, etc.). The Time column focuses on pertinent events and synchronization of operations. Finally, the Motivation category shows the rationale for the IS solution, and viewed vertically can be regarded as having ends-means-ends structure.

An attractive feature of this model is that it helps identifying the pertinent *gaps* to be addressed to completely describe any given system. Moreover, the dimensions of the framework seem to nicely fit the objectives of the present work: describing design researcher's artifact, in lights of incorporating relevant aspects and levels of refinement. In the subsequent sections we will modify the framework to encompass types of IS artifacts.

4. Dimensions of IS Representational Framework for Design Research

In this section we will introduce the framework for structuring a representation of IS design researcher's artifact. We will first consider the relevant perspectives of the framework. The issue is not only important for providing adequate reference models, but also for distinguishing different types of design research projects. For example, given a specific design research project, would it be appropriate to ask what the logical architecture and dynamic properties of a class of IS are? What are the technological platform choices? What is the computational complexity of the algorithms employed? Is the design of the system efficient? What are the implementation details? These questions seem to belong to different levels of IS representation, and must, therefore, be separated.

Re-examining Zachman's framework we could relate the planner and owner perspectives to compose what would be most descriptive as a layer that represents the IS from outside, i.e. essentially the requirements, or meta-requirements. We will refer to this layer as the "*analytical*" perspective on IS. The major purpose here is to represent the IS a set of relevant system characteristics and processes supported.

The analytical layer is of prime importance as it may serve as a point of convergence of behavioral and design studies. The behavioral research could incorporate the characteristics pertinent to the layer to theorize about potential impacts on organizational/individual performance. Design researchers could conceptualize novel features and functions within the context of the type of IS in question. Analytical layer erases sharp boundaries between description and design. For example, theorizing about active features of decision support systems (DSS) would lead to the description of the active DSS, and at the same time initiate the design of such systems.

Below the analytical layer there must be a layer that describes the organization of the artifact, i.e., its design. The "design" of IS that supports the analytical representation can be divided at least into two layers. The first of these focuses on the overall structure of the artifact to support the requirements of the system. In other words, the basic issues addressed here is how to synthesize an information system class from its components in order to achieve the "specifications" of the upper layer. The overall architecture and working principles of the system can be pictured here. We call this layer a *"synthetic"* perspective of the system.

The synthetic layer shows a way to employ the "knowledge base" from respective reference disciplines (Hevner et al. 2004), utilizing applicable concepts, methods, and system frameworks to compose an abstract system. It corresponds to the Designer (system model) view of Zachman's framework. It should not focus on the questions of the optimality, efficiency, ease of deployment and the like of the design of the proposed type of solution per se. Investigation of these issues is the focus of the *technological* layer, for the IS representation. This layer corresponds to what is usually regarded as "detailed design". However, the layer does not refer to concrete artifacts, but rather to more technical level descriptions of generic IS solutions. For example, if at the synthetic layer the system employs a search method, at the technological layer the question would be to investigate which search method promises the best performance in the context of the task that needs to be accomplished. In terms of Zachman's framework this layer would correspond to the builder's, and to some extent, subcontractor's views. Solutions devised at this layer should aim at providing informative guidance, a sort of generally defined specifications in order to implement concrete systems.

Finally, the last level of representation captures concrete system implementations. While research is primarily interested about general phenomena, the *implementation* layer serves as the ultimate connection with the real application environments. Design researchers in particular would like to see the results of their higher level artifacts to be implemented in practice. Metaphorically speaking the representations at this layer serve as *observations* to the generic constructs of the higher layers. Table 1 summarizes the proposed layers that belong to the "perspectives" dimension.

Table 1 Perspectives of the framework

Perspective	Description	Relationship to Zachman's framework
Analytical	Models salient system features and processes supported as characteristics of work/tasks performed within organizational/ human contexts	Planner, Owner (user)
Synthetic	Focuses on the organization of the system in terms of key capabilities of components to support the analytical representation	Designer
Technological	Focuses on the questions of the optimality and efficiency of the artifact's design	Builder, Subcontractor
Implementation	An instantiation of a system for a given concrete problem/ organization.	Functioning System

The second dimension deals with different categories or aspects that can be used to represent the type of IS being studied. In line with the generality requirement, we will be interested in capturing those categories that are appropriate for conceptualizing classes of IS, i.e. those conveying the essence of a given type of systems. The two essential categories include the ones that capture static and dynamic properties of the class of systems. Modeling of static and dynamic aspects has long been used in software engineering, where the focus is on providing tools and methods to produce concrete systems in an effective and efficient manner. The static (or structural) part of representation primarily corresponds to the "Data" view of Zachman's framework, while the dynamic description would most likely relate to the "Process" or "Function" as well as "Time" columns. These views allow the description of systems as structures and behaviors from each relevant perspective. For example, analytical layer could serve as a segment where the salient features as well as processes that describe business information systems are conceptualized.

Another important category derived from Zachman's framework is "Motivation". This view represents the system as the purpose (the function, the end) that it aims to fulfill at the given level of representation. As the development of specific systems is driven by specific business needs, the motivation behind classes of systems must be driven by the common needs and opportunities of the respective problem domain. Thus, for instance, the motivation behind software agent-based systems is to relieve humans of performing mundane tasks that are amenable to automation. Inclusion of this motivational category allows describing generic systems as purposes they fulfill. Thus, at any given level of representation one could obtain a full picture of how the static and dynamic properties of the systems relate to the major motivations behind them. In fact, for some artifacts (agents) the intentional characterizations are used explicitly in designing or theorizing about them (Wooldridge and Jennings 1995).

Since the "Network" component of Zachman's array seems to carry very much implementation-specific meaning, we decided to drop it from our representation. Furthermore, we will also omit the "People" column to emphasize the inseparability of human users and their tasks from the IS representation at the analytical layer. However, we feel that there is one more important category missing. While Zachman's framework shows various views of one concrete system instantiation, in our opinion each layer of our framework representing abstract artifacts can be complemented by specific instantiations, or prototypes. March and Smith have included instantiations as one of the outcomes of design research, mentioning that these could include both prototypes and implemented systems (March and Smith 1995). In our view it would be useful to provide instantiations at each layer of the representational framework that serves the purposes of illustrating, exemplifying, or implementing the contents of other cells of that layer.

Table 2 summarizes the four categories that can be employed to describe types of IS at any perspective layer.

Table 2. Categories of the framework

Perspective	**Description**	**Relationship to Zachman's framework**
Structure	Defines IS using structural and static descriptions, e.g. properties, subsystems, and relationships.	Data
Behavior	Shows dynamic aspect of IS, e.g. supported processes, working principles, and methods employed.	Function, Time
Motivation	Provides the description of IS from a given perspective as a set of primary motivations.	Motivation
Instantiation	Shows example instantiations for various perspectives, e.g. prototypes.	N/A

5. The matrix for IS representational framework

Similar to Zachman's model we can structure the researcher's representation for information systems using a two-dimensional matrix. Table 3 summarizes the framework in terms of dimensions introduced above.

The analytical layer represents the view of IS that not only informs the subsequent design decisions, but can also be used in behavioral studies to conceptualize the type of systems being investigated. The structure category allows the description of (generic) systems in terms of their salient features. The behavior category permits incorporating processes that are supported/exhibited by IS. The motivation category clarifies the description of the common business need addressed by the system. The Instantiation category allows to include an example representation of the system. In this aspect a mock-up prototype or verbal description would be equivalent to a high-level instantiation. A recent interesting work is well worth mentioning in this connection. Davis and Venkatesh have recently shown that pre-prototype testing can help predicting the perceived usefulness of implemented systems (Davis and Venkatesh 2004). Such a "pre-prototype" representation can be instantiated using descriptions and screenshots that accompany the questionnaire. An entry in the analytical/instantiation cell, could then be exemplified by employing such a "pre-prototype".

Table 3. A Representational framework for generic IS artifact

Category / Perspective	Structure	Behavior	Motivation	Instantiation
Analytical	Salient properties and features	Processes supported	Major generic business/ organizational motivations	Mock-up prototypes
Synthetic	Overall logical organization (architecture) of IS	The dynamic behavior of IS; its working principles	Motivation for the type of IS solution as related to requirements	Working prototype that illustrates the concept
Technological	Refined structural design for the type of IS	Refined methods, algorithms, interactions employed by IS	Objectives for the design	Generic packages, frameworks, shells.
Implementation	Architecture, design of implemented systems	Dynamic aspects of implemented systems	Specific business motivations	Implemented functioning systems

At the synthetic layer the design researcher would aim at synthesizing a type of the system solution that would conform to the requirements defined at the analytical layer. The major motivation here is to propose a solution that would accomplish this goal, while postponing the refinement of the design per se. The structural aspect would show an overall organization of the system in terms of its key components. The dynamic category would detail the behavior of the components, and interactions that take place among them, as well as with the user. Instantiation would contain a working prototype that exhibits the major essential features and processes supported. Such a prototype can demonstrate the feasibility of the solution, and also can be used in experimental assessments of the value of the concept.

The major motivations in the technological layer would most typically address the efficiency of solution, its technical generality, ease of development, optimality of technical choices, and the like. In a word the layer addresses the question what is the best technical design of the given system type. Thus, the architectural and behavioral aspects would be refined here, and the instantiations would ideally include frameworks, shells, patterns, components that could be used to efficiently build concrete systems. If we adopt the guideline suggested in (Hevner et al. 2004) to treat design as a search process, we could see that often the search for adequate conceptualization at the analytical layer could be complemented by the search for the synthetic model that supports the former. Essentially this corresponds to a search for a problem that itself is vaguely defined. Thus, until the value and validity of representations at the higher levels is supported by empirical evidence, it would be best to postpone the search for the adequate refined technological model.

As a brief example consider a recent publication describing an approach to support online shoppers in infrequent shopping tasks (Vahidov and Ji 2005).The work mostly addresses the analytical and synthetic conceptualizations. At the analytical layer the primary need addressed by the authors in this work is to help online shoppers make decisions in selecting products for infrequent purchases. On one hand, there could be multiple potential product offerings that could lead to cognitive overload experienced by the shoppers. On the other hand, the shoppers may not have well defined objectives, preferences, and criteria in making infrequent buying decisions. The authors primarily rely on principles adopted from the area of Human Problem Solving as the kernel theory. Authors emphasize "proactiveness" of their system that could serve as one of the salient features of their system's representation at the analytical layer. They further note that both searching and browsing capabilities need to be provided. However, the search should allow vague specification of the shoppers preferences, while browsing should be characterized as a divergent process, in line with problem solving principles. The analytical description of the system behavior can be summarized as: 1) vaguely defined search; 2) divergent product browsing; 3) convergent product browsing. The authors have chosen notebook selection as an instance of infrequent shopping task.

The analytical perspective provides a relatively technology-independent, and yet fairly descriptive representation of the system. It could be used not only to describe the requirements for design, but also to guide explanatory studies. For example, the above description could be used to investigate the impacts of divergent product recommendation on the effectiveness of product selection. However, for a design work, the researcher has to synthesize a solution that would exhibit the needed properties and behaviors. The synthetic structural representation of the system is provided in terms of its architecture including key parts and their relationships. The major parts include: user interface; recommendations generator; recommender profiles; criteria management; fuzzy filter; clustering module; and product database. The system behavior is specified in terms of the behavior of individual parts and their interactions. An online shopper uses a web-based interface to indicate his or her preferences using "fuzzy" terms, e.g. "The price is 'quite' to 'extremely' important". The filtering module then uses this information and fuzzy mathematics in order to partition products into different "classes" or "grades". The clustering algorithm then generates divergent recommendations for the shoppers, providing them with the capability to "zoom into" various regions of product space. The prototype system is presented as an instantiation at the synthetic layer.

While the current work does not elaborate on the methodology of design research, one point is worth mentioning here. It might appear that the natural progression through the layers for design researchers would be top-down, i.e. triggered by kernel theories, which may be described as *deductive* process. However, as the vast majority of innovative solutions are grounded in practical problems, and thus arise from the industry, the design research programs could be actually progressing from the bottom up. For example, proactive autonomous components had been in use before autonomous agents emerged as a research field. The issues facing the researchers then would be to investigate the applicability, generalizability, extensions and value of innovations in broader contexts. Such a process could be described as being essentially *inductive*. This might suggest that innovativeness, or originality is not as important requirement of design research as generality and rigor. However, in our view design research should attempt to provide a broader sort of innovations, the ones that are applicable in a variety of contexts and situations.

The point becomes clearer if one views implemented systems as sort of "observations" for higher layers. Thus, new particular "observations" could trigger the search for new theoretical models. As both "deductive" and "inductive" methods have been known for ages in scientific research, similar patterns seem to be legitimate approaches to conceptualizing systems at different layers in design research projects.

6. An Example of IS Representation: Situated Decision Support

Decision Support Systems constitute a general major type of Information Systems that has attracted many researchers' attention since the Seventies. The interest in using technology to inform organizational decision making had been triggered by specific requirements imposed by the nature of managerial work that demanded an approach significantly diverging from other types of systems (Alter 1980; Alter 1981; Keen and Morton 1978; Sprague and Carlson 1982). The early DSS concept could be described in terms of the representational framework proposed here. At the analytical layer, the major motivation behind DSS has been improving the effectiveness of human decision making through computerized means. In terms of salient characteristics, early DSS were envisaged as a sort of a "toolbox" aimed at supporting, not replacing decision makers, with emphasis on interactivity, user-friendliness, and adaptability. From the dynamic point DSS were conceptualized around Simon's problem solving (Intelligence-Design-Choice) model (Simon 1977). Alter provided a taxonomy to analytically distinguish different types of DSS based on the degree to which a system determines the decision, ranging from "File drawer systems" to "Suggestion Systems" (Alter 1980). This taxonomy offered a way to effectively characterize DSS in terms of their salient properties. At the synthetic layer, DSS was being pictured as consisting of models, data, knowledge, and user interface components. Technologically, specific systems could be assembled from DSS tools, or by means of "DSS generators" (Sprague and Carlson 1982).

Recent profound changes in information technologies offer new opportunities to enhance the traditional view of DSS conceptually and technologically. One of such proposed concepts is the notion of "situated" DSS, or "Decision Station" (Vahidov 2002; Vahidov and Kersten 2004). We will use this type of systems to illustrate how a representational framework can represent design researchers' IS artifact in a structured fashion.

Analytical Perspective

Motivation

The major motivation for the concept of situated DSS has been to bring the nature of decision support up to the levels of demands of the modern decision makers for supporting timely effective decisions in global, complex, and networked business environment. Traditional DSS have been criticized for being disconnected from problem domains and having toolbox nature, while the modern decision makers need means of continuously assessing developing problems and opportunities and responding to them in a timely fashion. The advance of the Internet and e-Business offers opportunities to develop a novel concept for a type decision support that would be closely linked to its problem environment. The motivation is also reinforced by comparing traditional Artificial Intelligence (e.g. Expert Systems) with Intelligent Agents. The latter could be seen as application of AI techniques to build situated systems. While the field of Intelligent Agents has gained much popularity and adoption, the authors argued that the field of DSS could also potentially benefit from moving to the concept of situated systems.

Structure

In elaborating on the salient features (essentially, meta-requirements) for the situated DSS, the authors' focused on those that are specific to these systems, thus we are assuming that the major characteristics of DSS in general (e.g. interactivity; decision support instead of automation) are being inherited as well. In this sense the situated DSS extends the concept of the traditional DSS. The two characteristics that are relevant for defining situatedness are direct links with the environment, and system proactiveness. While active DSS (actively supporting problem solving) has been one stream of research within the discipline (Angehrn 1993; Carlsson et al. 1999; Manheim 1988; Rao et al. 1994; Vahidov and Elrod 1999), proactiveness here relates to on-going interwoven interactions with the user, as well as with the environment. Both of these features have been adopted from the area of Intelligent Agents as means of facilitating "situatedness".

Behavior

Traditionally, Simon's three-phase (intelligence, design, choice) decision making model have been connected to the decision support. Situated DSS aims to target also the implementation and monitoring phases. Thus, decision making process can be viewed as an ongoing process of gathering important information, providing active support in formulating a decision, translating decision into actions, executing those actions, monitoring the outcomes, and alerting of possible problems and opportunities. The situated DSS, thus, makes a very strong emphasis on the dynamic aspects of decision making and implementation.

Instantiation

As an example illustration the authors have discussed the possibilities of employing situated DSS for investment management.

<u>Synthetic Perspective</u>

Motivation

The analytical perspective gives a relatively technology-independent, and yet fairly descriptive representation of the system. It could be used not only to describe the requirements for design, but also to guide explanatory studies. However, for a design work, the researcher has to synthesize a solution that would exhibit the needed properties and behaviors. Thus, a major motivation here is to provide an overall design for the type of the system that would support its analytical-layer representation.

Structure

The synthetic structural representation of the system was provided in terms of its architecture, including key parts and their relationships. The major parts included: sensors, effectors, active user interfaces, and DSS kernel. The primary function of the sensors is to provide timely information and alerts to other components of the system. The effectors aim at transforming the decisions made into actions and executing them. Both sensors and effectors are intended to be as close to the environment as the current technology and business practices permit. The sensors and effectors are the primary means of achieving the intimate links with the environment. They are not necessarily simple devices, but may have advanced capabilities, such as planning and adapting. The authors have also proposed the logical design for these components. The DSS kernel is envisioned as being composed of traditional DSS toolbox and an active "manager" that helps identifying relevant developing situations and execute or suggest corrections if necessary while using the toolbox. Active user interface aims to actively support the decision making process by the user while effectively hiding the complexities of the toolbox.

Behavior

While the authors have spent most of their efforts in formalizing the structural aspects of the system, the system behavior was described in an illustrative example of investment management. Generally speaking, the sensors are continuously (or periodically) observing the environment for significant changes in relevant variables. If detected the sensors could raise an alarm. The "manager" component then analyses the situation and if the problem is deemed to be minor, it may decide to take a corrective action autonomously. Otherwise, the user is notified and proactive decision support invoked by the active user interface. The decision, if made is then passed to effectors for the execution.

Instantiation

The prototype system for illustrating the concept using the investments area has been presented in the paper.

<u>Technological Perspective</u>

Since the purpose of the authors was to introduce a concept of situated decision support, they focused primarily on the analytical and synthetic aspects, while postponing the technological ones for later work. The major motivation for the refinement of the concept could be to offer a generic technical solution that would be relatively easy to adapt to a variety of infrequently purchased products. The adequateness of particular choices (structural and dynamic) and generality of the solution could be investigated. Some of the possible direction for further refinement could be specification of a generic solution using component- or service- oriented architectures. The structural aspects could be represented through a technical-level description of architecture (e.g. component diagram). The behavioral aspects would detail various possible interactions/exchanges among system components. The instantiation could include a set of components/services, a framework, or an adaptable packaged solution that could be employed to quickly implement a functioning system.

7. Conclusion

The major motivation for this work has been to propose a framework for representing types of information systems in design research. The framework could help design researchers to scope their efforts and conceptualize the structure and behavior of the types of systems they are working on. To this end we have adopted and adapted a framework originating from the industry to suit the design researcher's needs. The framework allows having an informative picture of an information system through various perspectives. In other words it answers (or, rather asks) the questions: What is the system as a structural entity? How does it behave (what processes does it exhibit/support/enhance)? Why does (or should) it exist (motivation, objectives)? Can an example be shown?

The framework allows conceptualizing artifact of various degrees of generality. For example, it could be used to represent higher classes of systems (e.g., DSS) as well as more narrowly defined ones. Thus, the framework is suitable for describing classes of systems of varying levels of generality, or abstraction. We believe that the representational framework introduced here will be helpful as a reference model in usefully informing design researchers, as well as scientists working on establishing the methodology of design research in information systems.

References

Albert, T.C., Goes, P.B., and Gupta, A. "GIST: a Model for Design and Management of Content and Interactivity of Customer-Centric Web Sites," *MIS Quarterly* (28:2) 2004, pp 161-182.

Alter, S. *Decision Support Systems: Current Practice and Continuing Challenges* Addison-Wesley, Reading, Mass., 1980.

Alter, S. "Transforming DSS Jargon into Principles for DSS Success," Transactions of the International Decision Support Conference, 1981.

Angehrn, A.A. "Computers that Criticize You: Stimulus-Based Decision Support Systems," *Interfaces* (23:3) 1993, pp 3-16.

Armour, F.J., Kaisler, S.H., and Liu, S.Y. "A Big Picture Look at Enterprise Architectures," *IEEE IT Professional* (1:1) 1999, pp 35-42.

Au, Y.A. "Design Science I: The Role of Design Science in Electronic Commerce Research," *Communications of the AIS* (7:1) 2001.

Ball, N.L. "Design Science II: The Impact of Design Science on Electronic Commerce Research and Practice," *Communications of the AIS* (7:2) 2001.

Burstein, F., and Gregor, S. "The Systems Development or Engineering Approach to Research in Information Systems: An Action Research Perspective," 10th Australasian Conference on Information Systems, Wellington, New Zealand, 1999, pp. 122-134.

Carlsson, C., Kokkonen, O., and Walden, P. "On the Improvement of Strategic Investment Decisions and Active Decision Support Systems," Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999.

Davis, F.D., and Venkatesh, V. "Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management," *IEEE Transactions on Engineering Management* (51:1) 2004, pp 31-46.

Hevner, A.R., March, S.T., Park, J., and Ram, S. "Design Science in Information Systems Research," *MIS Quarterly* (28:1) 2004, pp 75-105.

Keen, P.G.W., and Morton, M.S.S. *DSS: An Organizational Perspective* Addison-Wesley, Reading, 1978.

Kock, N., Gray, P., Hoving, R., Klein, H., Myers, M., and Rockart, J. "IS Research Relevance Revisited: Subtle Accomplishment, Unfulfilled Promise, or Serial Hypocrisy?," *Communications of the Association for Information Systems* (8) 2002, pp 330-346.

Livari, J. "The IS Core – VII: Towards Information Systems as a Science of Meta-Artifacts," *Communications of the Association for Information Systems* (12:37), November, 2003 2003, pp 568-581.

Manheim, M. "An Architecture for Active DSS," Proc. of the 21st Hawaiian International Conference on Systems Sciences, 1988, pp. 356-365.

March, S.T., and Smith, G.F. "Design and Natural Science Research on Information Technology," *Decision Support Systems* (15) 1995, pp 251-266.

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

Rao, H.R., Sridhar, R., and Narain, S. "An Active Intelligent Decision Support System," *Decision Support Systems* (12:1) 1994, pp 79-91.

Simon, H. *The Sciences of the Artificial*, (3 ed.) The MIT Press, Cambridge, Massachusetts, 1996.

Simon, H.A. *The New Science of Management Decision* Prentice-Hall, Englewood Cliffs, N.J., 1977, p. 175.

Sowa, J.F., and Zachman, J.A. "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal* (31:3) 1992, pp 590-616.

Sprague, R.H.J., and Carlson, E.D. *Building Effective Decision Support Systems* Prentice-Hall, Englewood Cliffs, NJ, 1982.

Vahidov, R. "A Notion for a Situated DSS," 35th Hawaii International Conference on System Sciences, 2002.

Vahidov, R., and Elrod, R. "Incorporating Critique and Argumentation in DSS," *Decision Support Systems* (26:3) 1999, pp 249-258.

Vahidov, R., and Ji, F. "A diversity-based method for infrequent purchase decision support in e-commerce," *Electronic Commerce Research and Applications* (4:2) 2005, pp 143-158.

Vahidov, R., and Kersten, G.E. "Decision Station: Situating Decision Support Systems," *Decision Support Systems* (38:2) 2004, pp 283-303.

Walls, J.G., Widmeyer, G.R., and El Sawy, O.A. "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research* (3:1), March, 1992 1992, pp 36-59.

Wooldridge, M., and Jennings, N. "Intelligent Agents: Theory and Practice," *Knowledge Engineering Review* (10:2) 1995.
Zachman, J.A. "A Framework for Information Systems Architecture," *IBM Systems Journal* (26:3) 1987, pp 276-292.