Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2008 Proceedings

International Conference on Information Systems (ICIS)

2008

# Conceptualizing the IT Artifact for MIS Research

Sabine Matook
*The University of Queensland*, s.matook@business.uq.edu.au

Susan A. Brown
*The University of Arizona*, suebrown@eller.arizona.edu

# Conceptualizing the IT Artifact for MIS Research

*Conceptualisation de l'artéfact informatique pour les recherches en management des systèmes d'information*

*Research-in-Progress*

**Sabine Matook**
The University of Queensland
Business School
3 Blair Drive
Brisbane, Australia
s.matook@business.uq.edu.au

**Susan A. Brown**
The University of Arizona
Eller College of Management
MIS Department
1130 East Helen St.
Tucson, Arizona 85721
suebrown@eller.arizona.edu

## Abstract

*The notion of the information technology (IT) artifact has received a great deal of attention, particularly since Benbasat and Zmud's (2003) call for it to be the core of the information systems discipline. Yet, little work has focused on defining and discussing the IT artifact in a way that can facilitate consistent treatment across studies. In this paper, we develop a taxonomy of the IT artifact. The taxonomy is derived from literature on general systems theory and Akerman and Tyree's (2006) architectural ontology. We provide a preliminary explication of the taxonomy using four different systems as examples. We also discuss the iterative approach we will take to develop the taxonomy more completely. Our objective is to develop a taxonomy that will provide a language for IS researchers to use when discussing the IT artifact.*

**Keywords:** IT artifact, taxonomy, ontology, general systems theory

## *Résumé*

*Récemment, la notion d'artéfact informatique a fait l'objet d'une attention particulière (Benbasat et Zmud, 2003). Bien que plusieurs discussions aient été engagées, peu de travaux se sont intéressés à la définition-même de l'artéfact informatique. Dans cette recherche nous en proposons une taxonomie fondée sur la théorie générale des systèmes et l'ontologie architecturale de Akerman et Tyree (2006).*


## Introduction

Benbasat and Zmud (2003) initiated a somewhat controversial exchange in which they put forth a call to have the information technology (IT) artifact be the core focus of research in information systems (IS). A number of authors responded to this call, suggesting, for example, that while the artifact is important, so are the structure, context, and other aspects of research that examine the artifact (e.g., DeSanctis 2003), and that the IS discipline ought not define itself too narrowly (e.g., Galliers 2003; Robey 2003). Benbasat and Zmud's article spawned a great deal of discussion regarding whether or not the IT artifact is, in fact, the core of the IS discipline, but very little discussion regarding the IT artifact itself.

A cursory review of the literature reveals a wide range of perspectives on the IT artifact[1], with some research identifying a very particular application, such as a portfolio analyzer for financial analysts (Venkatesh et al. 2003), and other research studying electronic commerce adoption (Pavlou and Fygenson 2006). The level of analysis varies significantly across these types of systems with the former designed for a specific task and the latter potentially encompassing multiple tasks. In their extensive review of the literature, Venkatesh et al. (2003) indicated that prior adoption research examined simple, individual-oriented, stand-alone systems as opposed to the more complex and sophisticated systems we see in organizations, and that this would have implications for interpreting findings across studies. This highlights the importance of clearly and consistently conceptualizing the IT artifact in our research.

General Systems Theory (GST: von Bertalanffy 1968) provides a theoretical foundation to define the real world in terms of systems and sub-systems that embrace interlinked elements. This is consistent with work in classification that is concerned with a systemic and sound description and differentiation of real world constructs by a certain set of characteristics (Bailey 1994). Through GST, knowledge in a certain segment of the world can be captured, explained and predicted at different levels of abstraction. A similar objective is pursued in research on ontologies. Studies on ontologies seek to define fundamental things that make up the world as well as the relationships among those things. Rooted in philosophy, ontologies focus on the structure and the behavior of the world as we, as individuals, perceive it. In so doing, the ontology can be, and has been, used to represent the IT artifact in various facets (Wand et al. 1999). The challenge with the ontology is to bridge the gap between the abstract ontological representation of the artifact and the real IT artifact. In particular, the Bunge-Wand-Weber (Wand and Weber 1990, 1993, 1995) ontology used concepts like *thing, property* and *composition of things* for their conceptualization. We propose to close the gap between these concepts and the actual IT artifact.

Given this backdrop, this paper attempts to develop a taxonomy for discussing the IT artifact in IS research. Our objective is to augment existing ontologies using general systems theory to arrive at a taxonomy that can be leveraged in IS research. Our ultimate goal is to provide a means by which IS researchers can theorize about the IT artifact in a consistent manner, thus enabling results across studies to be compared based, in part, on key characteristics of the IT artifact in question.

## Theoretical Background

In this section we first define the components of an IT artifact by drawing on general systems theory and ontologies. Then, we present a set of system characteristics that explain components of the IT artifact, independently of the context in which it is implemented.

[1] We focus on the IT artifact as software application; we acknowledge that hardware and development methodologies can also be considered IT artifacts, but we do not address them in this paper.

### *Defining Components of the IT Artifact*

Research in IS addresses questions of adopting, implementing, using, and building the IT artifact. While multiple definitions of the IT artifact exist, we leverage the one proposed by Benbasat and Zmud (2003): “the application of IT to enable or support some task(s) embedded within a structure(s) that itself is embedded within a context(s)” (p. 186).

Basically, the IT artifact serves as the centralizing theme, or identity, for the field (Benbasat and Zmud 2003). Given the importance of the IT artifact, we would assume that IS research would include it without exception. Yet, in their analysis of the papers published in *MIS Quarterly* and *Information Systems Research* in the years 2001 and 2002, Benbasat and Zmud (2003) found that about 33% of the papers neither focus on the IT artifact nor include related issues like IT capabilities, IT practice, IT usage, or IT impact. Similarly, in their paper on theorizing the IT artifact, Orlikowski and Iacono (2001) found that 22% of papers published in *Information Systems Research* between 1990 and 1999 excluded the IT artifact. Their study offers five mutually exclusive views on technology, namely tool, proxy, ensemble, computational, and nominal. While this is an interesting perspective, it does not provide a means of clearly differentiating across IT artifacts. In essence, it is still too broad to be of use in IS research.

Both Benbasat and Zmud (2003) and Orlikowski and Iacono (2001) discuss the IT artifact in a general manner and do not differentiate across types of artifacts. For example, Benbasat and Zmud (2003) present budget planning software and Internet presence as two examples of the IT artifact. We argue that these two IT artifacts are quite different. They represent different goals and are at very different levels of analysis, with Internet presence being too broad to provide meaningful differentiation in terms of theorizing. While there is a call to theorize the IT artifact, efforts to date have not provided a meaningful mechanism for doing so.

General systems theory (GST: van Bertalanffy 1968) offers one theoretical perspective for defining the IT artifact. GST indicates that systems are arrangements of certain elements that are interrelated in such a way that they form a whole (Klir 1972). Each system has at least two interconnected elements that are surrounded by system boundaries. The boundaries separate the elements from their system environment (Boulding 1956). Although the system boundaries are objective entities, they show a notion of subjectivity insofar as the point of view and interest of the observing individual impacts the boundary definition (Ackoff 1971). The relationships among the elements support connectivity and also enable them to bridge the system boundaries, via interfaces, to ensure the elements are responsive toward environmental changes. The relationships also explain how the system elements interact and fit together. The elements can form sub-systems, which are connected either directly or indirectly to every other sub-system or single element (Ackoff 1971). Both the sub-systems and the single elements aim for purposeful interoperation (Kast and Rosenzweig 1972). The purposeful behavior of systems manifests in the overall system goal and further in the sub-system goals. GST proposes that the reason for system creation is goal fulfillment.

Each system element embraces a unique set of characteristics that support the comparison, differentiation, and combination among the elements. The characteristics that emerged out of the theoretical work of GST are closed-open and static-dynamic (Boulding 1956, van Bertalanffy 1968, Ackoff 1971, Kast and Rosenzweig 1972). The closed-open characteristic translates into related characteristics that describe how the elements communicate (synchronous-asynchronous) and how the elements are connected (integrated-stand-alone). The static-dynamic characteristic is concerned with the system’s capability to change, how the system deals with the change (adaptive-non-adaptive), and the required state of the system to respond to changes (stateless-stateful). These characteristics, and their interactions, enable the definition of different types of systems, such as social, mechanical, and biological systems, and any combination of them. Figure 1 illustrates GST.

Another theoretical perspective, consistent with GST, for defining the IT artifact is ontologies. Within the software engineering literature, ontologies (e.g., Akerman and Tyree 2006) have emerged as a popular means of expressing system components. Ontologies have a philosophical origin and can be thought of as a systematic account for “existence”. The logic is that only what exists can be represented. Ontologies are explicit formal specifications of a shared conceptualization (Gruber 1993). A conceptualization is an abstract, and hence simple, view of the world we want to represent. It is the mental process where the individual forms his/ her understanding about a part of the reality. Computer science uses ontologies to achieve consensus about the meaning of the things in this reality and hence, to have a common understanding of them (Sanchez et al. 2006). Similar to GST, ontologies describe the order and structure of reality independent of the perspective and any other influence of the observer (Shanks et al. 2003). Thus, they can be applied to identify the objects and relationships among them in a given domain (Wand and Weber 2004). If the domain changes, then so do the objects and the relationships (Sanchez et al. 2007).

The literature provides numerous ontologies that have been created with different purposes in mind for different domains (e.g., Osterwalder et al. 2005; Ushold and King 1995; Ushold et al. 1998; Green and Rosemann 2005). Osterwalder et al. (2005) created an ontology to clarify the concept of business models, their uses, and their roles in the IS domain by defining nine building blocks that embrace the business model components. This ontology was also applied to the domain of e-business where it was used to conceptualize e-business models into elements, relationships, vocabulary, and semantics of the essential subjects (Osterwalde and Pigneur 2002). Ushold et al. (1998) and Ushold and King (1995) developed an enterprise ontology that includes terms and definitions that can be used to describe business enterprises, such as activity, sub-activity, activity owner, and also for sale and sale offer (Ushold et al. 1998, p. 40). In the field of system analysis and design Green and Rosemann (2005) use the Bunge-Wand-Weber ontology (Bunge 1977, 1997; Weber 1997) to investigate modeling techniques such as UML. In so doing, they identify the weaknesses of certain techniques and provide advice on their use for practice.

In addition to having ontologies for different purposes, the ontologies also differ in their levels of abstraction and proximity to the IT artifact. At a higher level, ontologies articulate fundamental ways of representing a general, real-world phenomenon. For example, Wand and Weber (2002) examine the capability of conceptual-modeling grammars to represent the IT artifact as a real-world phenomenon. At an intermediary level, an ontology describes the IT artifact in a particular domain, such as the ontology by Hajdukiewicz et al. (2001) for medical informatics. At a lower level of abstraction, ontologies describe a specific world from a particular point of view. For example, Timpf (2002) presents an ontology of a public transportation system from the user's perspective and the designer perspective. None of these examples helps us to clearly and consistently talk about the IT artifact.

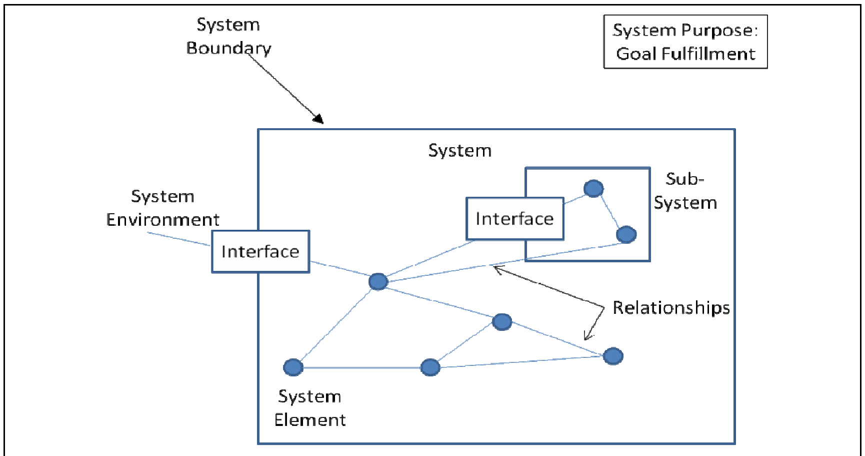


**Figure 1. General System Theory**

Of particular interest for our research is an ontology that is close to the IT artifact, namely an architectural ontology that is used to support the development of software architectures (Akerman and Tyree 2006). This type of ontology describes the domain by defining required components and relationships of the domain. The ontology offers four segments: (1) architecture assets, e.g. sub-systems, elements, and interfaces, (2) architectural decisions, (3) stakeholder concerns, and (4) architecture roadmap (how to build the architecture). The ontology aims to link stakeholders to the components (i.e., architecture assets) through architectural decisions that provide an architectural roadmap for development. In their ontology, Akerman and Tyree propose that a system is composed of sub-systems that aggregate elements and use data by deploying nodes (printers, PC) which are offered by interfaces. Consistent with GST, Akerman and Tyree's work emphasizes the partitioning of a system into sub-systems, which again contain smaller units, the elements. Indeed, their ontology and GST focus on representing the system, the embedded components and the relationships among them to create a common understanding of the domain.

Prior research has examined a variety of different systems, such as management support systems (e.g., Clark et al. 2007), health information systems (e.g., Braa et al. 2007), and enterprise systems (e.g., Ward et al. 2005). While this research is important, the systems studied are characterized at such a high level that it becomes difficult to look across studies to identify commonalities and differences. In IS research in general, little attention is paid to characteristics that define the systems under investigation and that would allow differentiating the IT artifacts in these studies. Only a few studies describe characteristics of the IT artifact, and these studies tend to focus on a particular characteristic. For example Leifer (1988) categorized computer-based IS into centralized, distributed, decentralized, and stand-alone system. A comprehensive set of characteristics that is general enough to apply to a broad range of IT artifacts is missing in the literature; yet given the call to theorize the IT artifact, it is sorely needed. The challenge when examining the IT artifact is to find the optimum degree of abstraction. A taxonomy that is too specific provides little common ground for comparison, while one that is too general has limited value in singling out differences. By integrating the ideas from ontologies with the theoretical viewpoint of GST we propose to offer a set of characteristics for conceptualizing the IT artifact. Ontologies focus on the system elements and their characteristics, while GST develops a clear separation between system elements and the environment by describing system boundaries. Thus, the integration provides a mechanism for developing a taxonomy that can be used for a variety of systems at a variety of levels. The next section presents our taxonomy.

## *Defining Characteristics of the IT Artifact*

Through the integration of GST and the Akerman and Tyree (2006) ontology, we follow a hierarchical, top-down approach to classifying the IT artifact. This approach helps to identify the levels of analysis inherent in different system views and provides a comprehensive representation of the IT artifact components. The artifact can be viewed from different levels of abstraction, but the main focus is on the system and its sub-systems. Through the relationships, which connect the system elements to each other and also to the environment, the IT artifact implements the system characteristics. The system goal defines the overall objective of the systems and other subordinated goals. The functionality of the system relates to the overall goal. The Akerman and Tyree (2006) ontology conceptualizes the interface as a service of the system. We extend the notion of service in such a way that it represents the interface to the system user – the individual who benefits from the system's goal achievement.

The goal of the system can be related to the task the users want to perform with it (Goodhue and Thompson 1995). The task emerges in the environment of the IT artifact. While the architectural decisions should be made to support the completion of the task, they also address the soft goals associated with the concerns of the stakeholders. These concerns are typically with respect to the IT artifact's capabilities to be of high quality, low risk for the firm, and responsive to the business needs. They, of course, then translate back to the design of the architecture assets. All encompassing is the roadmap that is interwoven among the concerns and the resulting implications, which then provide direction on migrating from the current system to the target system (Akerman and Tyree, 2006).

Prior research has posited different system characteristics that can be associated with each of the levels discussed in GST. A representative list is presented in Table 1, including sample references to help clarify the concepts. The definitions are taken from the online Oxford Dictionary of Computing and from the references cited.

We do not propose, nor do we believe, that the list is exhaustive. Instead, it provides a starting point for exploring the various systems and sub-systems that may prove meaningful for conceptualizing the IT artifact. Further, we expect that, in practice, certain combinations of these characteristics occur more often than others. We also expect that certain characteristics' combinations are meaningful, while others are necessarily mutually exclusive. For example, it is very unlikely that an IT artifact communicates synchronously but is implemented as a purely standalone system. In the following sections, we describe the five IT artifact characteristics.

### Synchronous – Asynchronous

The time aspect of communication and message transfer between IT artifacts is a criterion that enables differentiation, namely synchronous and asynchronous (DeSanctis and Gallupe 1987). The different transfer mechanisms impact organizational effectiveness and performance. With a synchronous system, both the sender and receiver have to be available at the same time, while in an asynchronous form the parties can access information on their individual schedules. Asynchronous communication indicates independence of one IT artifact from another IT artifact because no constant communication stream needs to be established. Email systems are examples of asynchronous communication, while instant messaging systems are examples of synchronous communication systems (Nah et al. 2002). Ngwenya and Keim (2003) investigated synchronous group decision support systems (sGSS) and asynchronous group decision support systems (asGSS) and found that asGSS facilitate higher decision quality than sGSS. The sGSS created time pressure for the group and unnecessary information overload. Similarly, Palvia (2001) investigated the use of learning systems in a synchronous mode during class and asynchronous outside of class. The results indicated that students did not have a preference for the synchronous support. These findings provide evidence that distinguishing systems on these characteristics is valuable for interpreting research results.

**Integrated – Standalone**

The level of integration of the IT artifact into the business environment has implications on data security, connectivity, and responsiveness. Integrated systems have more organizational impact than stand-alone systems (Noble 1993), e.g. in reducing the impact of imbalances in capacity and inventory planning (Caggiano et al. 2006). Stand-alone IT artifacts are completely independent and require no relationships with other systems to support their operation. They are implemented as centralized single-systems. The emphasis with stand-alone implementation is on user-friendly interfaces (Chou 1998). In contrast, integrated IT artifacts are part of a network and require support for their operation. Examples for integrated IT artifacts are client-server implementations and data-warehouse solutions (March and Hevner 2005). The benefit of integrated systems is the access to various information resources that they enable. Many organizations have experienced significant challenges in implementing integrated systems (e.g., Janssen and Cresswell 2005; Brown and Vessey 1999), and Brown et al. (2003) identify degree of integration as a key component in assessing how mandatory a system is. Thus, accounting for the degree of integration associated with a system or sub-system is valuable for interpreting research results.

**Table 1. Characteristics of the IT artifact**

| Characteristic | Definition | Relevant Studies |
|---|---|---|
| Static | Not changing or unable to take place during some period of time, usually while a system is in operation or a program is running | Bianchini et al. (2006), Rainer (1996), Gu et al. (2007), Hamadi and Benatallah (2004), vanderAalst (2000) |
| Dynamic | Capable of changing or of being changed, change is constant over a time period, for an operating systems, the implication is that the system is capable of changing while it continues to run | |
| Adaptive | System can adjust to varying levels of requests and interaction, e.g. user skills, system parameters are automatically adjusted as conditions change so as to optimize performance | Chuang and Yadav (1998), Deng and Chaudhury (1992), Fazlollahi et al. (1997) , Lau et al. (2008), Yu et al. (2003) |
| Non-adaptive | System cannot adjust for change in response to variations of interactions | |
| Synchronous | Sequential events take place at fixed times, transmission to a receiver happens instantaneously, it requires no acknowledgement that preceding events have been completed | Nah et al. (2002), Ngwenya and Keim (2003), Palvia (2001), Spencer and Hiltz (2003) |
| Asynchronous | Specific operation is begun upon receipt of an indication (signal) that the preceding operation has been completed, and which indicates to a subsequent operation when it may begin. | |
| Integrated | Interaction with the connected systems parts and the environment, combines some of their functions | Caggiano et al. (2006), Chou (1998), March and Hevner (2007), Janssen and Cresswell (2005), Leifer (1988) |
| Standalone | Has no interaction and connection to the environment, system is able to operate independently of other hardware or software parts | |
| Stateless | Without a state, system does not remember preceding sequential events or requests among elements, after a service request has been completed, it discards the data associated with the request | Osrae et al. (2007), Schulzrinne (1996), Thomas (1997) |
| Stateful | Full of state, system maintains state and keeps a record of the state of interaction, data about previous requests for a service which can then be used for subsequent requests | |

**Adaptive – Non-adaptive**

The capability of the IT artifact to adapt to changes in the environment (Chuang and Yadav 1998) and to be responsive is an additional differentiating characteristic. Advanced search engines, knowledge-based systems, and learning systems can adapt to changes either by improving their internal elements or by creating new relationships among elements. The capability of a system to learn means that the system is enabled to perform the same task effectively the next time it is initiated (Deng and Chaudhury 1992). This characteristic is particularly important in a globalized business world that uses Internet technologies to respond in an agile and flexible manner to changing customer demands and competitors threats. For example, adaptive negotiation agents in e-marketplaces learn to negotiate and thus, enhance the efficiency of the marketplace (Lau et al. 2008). In elderly care, adaptive agents (i.e., IT artifacts) can respond to the mental and physical capabilities of elderly people and then adapt to them (Yu et al. 2003). The ability to adapt is an important characteristic in differentiating across IT artifacts.

**Static – Dynamic**

Static systems undergo minimal changes over time. An example would be typical back-end applications in banks. They have minimal flexibility and neither scale nor any extension of functionality is possible (Ackoff 1971). In contrast, various online applications, for example for accommodations, airline bookings, online banking systems, and also web services (vanderMeer et al. 2003), are constantly changing to address the dynamics of the environment (Gu et al. 2007). Those systems are scaling to handle more users with the same system performance. In doing so, customer service is improved and the system enhances functionality as well as usability. The dynamic characteristic is also important for workflow management systems to cope with expected, but unusual situations of business processes (Hamadi and Benatallah 2004, vanderAalst 2000). Real time interaction between two systems is another criterion of dynamic systems as demonstrated for dynamic telepathology (Rainer et al. 1996). The degree to which systems are static versus dynamic thus seems to have implications for theorizing.

**Stateless – Stateful**

This characteristic relates to the systems capability to maintain state or not. Stateful systems remember previously performed events in a given sequence of interaction between system elements because the system stores state information. The next time the system is used, these data are activated to ease the usage. Stateless systems do not keep track of any state information and for each interaction with the system state information is requested (Fowler 2003). For example, the numerous html pages of the www make it a stateless system because no record is kept of general web-surfing interactions (Schulzrinne 1996). Stateless systems have the advantage that they can respond without preceding status information that would occupy enormous storage capacity (Osrae et al. 2007). However, in certain situations in particular e-commerce shopping applications, stateful systems are preferred, for example to identify customers and to implement shopping carts. In the Internet environment, stateful systems can be achieved with cookies (Thomas 1997). Whether or not a system has a record of prior transactions can be a significant factor in how the system interfaces with its environment as well as other systems.

### *Application of the IT Artifact Characteristics for selected Information Systems*

This section presents a preliminary application of the taxonomy to classify different IT artifacts. We show that systems with the same system goal embody different system characteristics. Before we do examine systems that are at a similar level of analysis, we want to demonstrate how to navigate through the levels. We use the example of the Internet for this purpose. The system goal is "to provide a common network to facilitate transfer and communication" (Turban et al. 2006) and the system embraces a number of sub-systems; e.g., the web, e-mail, and FTP. If we take the web as our sub-system, we can identify these elements (as examples): HTML pages in general, input/output screen, and hyperlinks. Instances of a web system would be search engines like Google or Yahoo. The framework can be applied across the different levels and in the Internet example differences of the characteristics at different levels exists. For example, an FTP server is stateful but the HTML web pages are stateless.

The application of the taxonomy is demonstrated for management support systems (MSS) as presented by Clark et al. (2007). The system would be MSS, while the sub-systems include decision support systems (DSS), executive information systems (EIS), knowledge management systems (KMS), and business intelligence (BI). Each sub-system has a common goal: to support managerial actions and "improved decision making effectiveness" (Clark et al. 2007, p.582).

DSS are static in nature and are primarily non-adaptive systems. The system architecture aims to provide a reliable decision aid. Interaction with a DSS tends to occur in a synchronous manner. The systems tend to be stateless – basing each new interaction on only the information available at the moment. Interestingly, a DSS can be both standalone and also integrated into a larger system landscape, as in a DSS associated with a data warehouse. EIS can be classified similarly, except that EIS are always integrated systems to enable a decision process that uses hard and soft data (Watson et al. 1996).

KMS and BI are dynamic systems that respond and adapt to changes in the system environment. It is important that these systems can be updated with new data and thus incorporate less codified data, and respond flexibly to the changes in the system environment. Both KMS and BI are stateful and transmit information synchronously. While KMS can be implemented as standalone and integrated, BI systems require the latter. The characteristics of BI systems support the observation that users are overwhelmed by information overload (Clark et al. 2007).

It is interesting to note that none of the four systems has the same set of characteristics. Table 2 summarizes the characteristics for the four sub-systems.

**Table 2. Application of IT artifact characteristics on Management Support Systems (Clark et al. 2007)**

| Characteristic | DSS | EIS | KMS | BI |
|---|---|---|---|---|
| Static – Dynamic | Static | Static | Dynamic | Dynamic |
| Adaptive – Non-adaptive | Non-Adaptive | Non-Adaptive | Adaptive | Adaptive |
| Synchronous – Asynchronous | Synchronous | Synchronous | Synchronous | Synchronous |
| Integrated – Standalone | Standalone - Integrated | Integrated | Standalone - Integrated | Integrated |
| Stateless – Stateful | Stateless | Stateless | Stateful | Stateful |
| System Goal | support managerial actions &decision making | | | |

## Method

In order to augment and provide validation for our taxonomy, we will survey the publications from 2006 and 2007 in the AIS list of six journals (*European Journal of Information Systems*, *Information Systems Journal*, *Information Systems Research*, *Journal of Association of Information Systems*, *Journal of Management Information Systems*, *Management Information Systems (MIS) Quarterly*). We will identify the IT artifact(s) studied based on the authors' descriptions. We will then map the characteristics of the IT artifact using the characteristics in our taxonomy and their definitions. As differentiating characteristics emerge, we will augment the taxonomy, iterating through the papers we have already examined. The definition of a set of IT artifact characteristics is necessarily an iterative process that allows us to alter our compilation and enrich it with characteristics we have seen in the papers analysed. Given some of the examples we provide above, we anticipate that that there may be insufficient details regarding the IT artifact to classify them. In that case, we will attempt to classify the artifact based on generally accepted definitions and interactions with the authors. Finally, we will share the taxonomy with colleagues in order to obtain external assessment of its applicability in ongoing research.

By the time of the conference, we will have completed the coding of the IT artifact for papers published in 2006 and 2007. We will also have augmented our taxonomy by mapping the artifacts onto it. Finally, prior to the conference, we will share the taxonomy with colleagues to obtain external validation.

## Expected Contribution

As stated in the introduction, our ultimate goal is to provide a means by which IS researchers can theorize about the IT artifact in a consistent manner, thus enabling results across studies to be compared based, in part, on key characteristics of the IT artifact in question. By developing a taxonomy that is broadly applicable at the system, sub-system, and element levels we provide a means of distinguishing across system levels. In addition, the taxonomy will enable development of a common treatment and understanding of the IT artifact in IS research. As an example, the taxonomy should enable researchers to assess the similarities across multiple studies that examine BI systems.

## Conclusion

In this research in progress, we present a taxonomy for classifying the IT artifact and theorizing about it in IS research. The taxonomy integrates GST and the Akerman and Tyree (2006) ontology. We identify a number of characteristics that derive from this integration. Examples from the literature that coincide with each characteristic are provided. A preliminary explication of the taxonomy is provided by comparing DSS, EIS, KMS, and BI systems.

## References

Ackoff, R. "Towards a system of systems concepts," *Management Science* (17:11), 1971, pp. 661-671.

Akerman, A., and Tyree, J. "Using ontology to support development of software architectures," *IBM Systems Journal* (45:4), 2006, pp. 813-826.

Bailey, K.D. *Typologies and Taxonomies: An Introduction to Classification Techniques*, Sage Publications, Thousand Oaks, CA, 1994.

Benbasat, I., and Zmud, R. "The identity crisis within the IS discipline: Defining and communicating the discipline's core properties," *MIS Quarterly* (27:2), 2003, pp. 183-194.

Bianchini, D., De Antolellis, V., Pernici, B., and Plebani, P. "Ontology-based methodology for e-service discovery," *Information Systems* (31:4-5), 2006, pp. 361-380.

Braa, J., Hanseth, O., and Heywood, A. "Developing health information systems in developing countries: The flexible standards strategy," *MIS Quarterly* (31:2), 2007, pp. 381-402.

Brown, C., and Vessey, I. "ERP implementation approaches: toward a contingency framework," in *Proceedings of the 20th international conference on Information Systems*. Association for Information Systems, Atlanta, GA, USA, December 1999, pp. 411-416.

Boulding, K. "General Systems Theory: The Skeleton of Science," *Management Science* (2:3), 1956, pp. 197-208.

Brown, S.A., Massey, A.P., Montoya-Weiss, M.M., and Burkman, J.R. "Do I really have to? User acceptance of mandated technology," *European Journal of Information Systems* (11:4), 2002, pp. 283-295.

Bunge, M. *Treatise on Basic Philosophy: Vol. 3: Ontology I: The Furniture of the World*. Boston: Reidel, 1977.

Bunge, M. *Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems*. Boston: Reidel, 1979.

Burton-Jones, A., and Straub, D. "Reconceptualizing system usage: an approach and empirical test," *Information Systems Research* (17:3), 2006, pp. 228-246.

Caggiano, K., Muckstadt, J., and Rappold, J. "Integrated real-time capacity and inventory allocation for reparable service parts in a two-echelon supply system," *Manufacturing & Service Operations Management* (8:3), 2006, pp. 292-319.

Chan, S., and Ngai, E. "A qualitative study of information technology adoption: how ten organizations adopted Web-based training," *Information Systems Journal* (17:3), 2007, pp. 289-315.

Chou, S. "Migrating to the Web: A Web financial information system server," *Decision Support Systems* (23:1), 1998, pp. 29-40.

Chuang, T., and Yadav, S. "The development of an adaptive decision support system," *Decision Support Systems* (24:2), 1998, pp. 73-87.

Clark, T., Jones, M. and Armstrong, C. "The dynamic structure of management support systems: Theory development, research focus, and direction," *MIS Quarterly* (31:3), 2007, pp. 579-615.

Deng P. and Chaudhury A. "A conceptual model of adaptive knowledge-based systems," *Information Systems Research* (3:2), 1992, pp. 127-149.

Dennis, A., Valacich, J., Fuller, M., and Schneider, C. "Research standards for promotion and tenure in Information Systems," *MIS Quarterly* (30:1), 2006, pp.1-12.

DeSanctis, G. "The social life of information systems research: A response to Benbasat and Zmud's call for returning to the IT artifact," *Journal of the Association for Information Systems* (4:16), 1003, pp. 360-376.

DeSanctis, G., and Gallupe, R. "A foundation for the study of group decision support systems," *Management Science* (33:5), 1987, pp.589-609.

Fazlollahi, B., Parikh, M., and Verma, S. "Adaptive decision support systems," *Decision Support Systems* (20:4), 1997, pp. 297-315.

Fowler, M. *Patterns of Enterprise Application Architecture*. Addison-Wesley, Boston, 2003.

Galliers, R. D. "Change as crisis or growth? Toward a trans-disciplinary view of information systems as a field of study: A response to Benbasat and Zmud's call for returning to the IT artifact," *Journal of the Association for Information Systems* (4:6), 2003, pp. 337-351.

Goodhue, D., and Thompson, R. "Task-technology fit and individual performance," *MIS Quarterly* (19:2), 1995, pp. 213-236.
Green, P., and Rosemann, M. "Applying ontologies to business and systems modeling techniques and perspectives: Lessons learned," *Journal of Database Management* (15:2), 2005, pp. 106-118.
Gruber, T. "A translation approach to portable ontology specifications," *Knowledge Acquisition* (5:2), 1993, pp. 99-220.
Gu, N., Cui, J., Ye, W., Wang, H., and Pei, J. "A system framework for web service semantic and automatic orchestration", in *Proceedings of the 2nd International Conference on Pervasive Computing and Applications, 2007*, pp. 606-11.
Hajdukiewicz, J.R., Vicente, K.J., Doyle, D.J., Milgram, P., and Burns, C.M. "Modeling a medical environment: An ontology an ontology for integrated medical informatics design," *International Journal of Medical Informatics* (62:1), 2001, pp. 79-99.
Hamadi, R., and Benatallah, B. "Recovery nets: Towards self-adaptive workflow systems," in *Proceedings of the International Conference on Web Information Systems Engineering (WISE)*, LNCS 3306 (2004), pp. 439-453.
Janssen M., and Cresswell , A. "An enterprise application integration methodology for e-government," *Journal of Enterprise Information Management* (18:5), 2005, pp. 531-547.
Kast, F., and Rosenzweig, J. "General systems theory: Applications for organization and management," *The Academy of Management Journal* (15:4), 1972, pp. 447-465.
Klir, G. *Trends in general systems theory*. Wiley-Interscience, New York, 1972.
Lau, R., Li, Y., Song, D., and Kwok, R. "Knowledge discovery for adaptive negotiation agents in e-marketplaces," *Decision Support Systems* (45:2), 2008, pp.310-323.
Leifer, R. "Matching computer-based information systems with organizational structures," *MIS Quarterly* (12:1), 1988, pp. 63-73.
March S., and Hevner, A. "Integrated decision support systems: A data warehousing perspective," *Decision Support Systems* (43:3), 2007, pp. 1031-1043.
Markus, M., Majchrzak, A., and Gasser, L. "A design theory for systems that support emergent knowledge processes," *MIS Quarterly* (26:3), 2002, pp.179-212.
Nah, F., Siau, K., Tian, Y., and Ling, M. "Knowledge management mechanisms in e-commerce: A study of online retailing and auction sites," *Journal of Computer Information Systems* (42:5), 2002, pp. 19-28.
Ngwenya, J., and Keim, R. "The effects of augmenting face-to-face meetings with web-based asynchronous group support systems," *Journal of Information Technology Theory and Application* (5:2), 2003, pp. 47-62.
Noble, F. "Implementation strategies for office systems," *Journal of Strategic Information Systems* (4:3), 1995, pp. 239-253.
Palvia, S. "Effectiveness of asynchronous and synchronous modes for learning computer software for end users: an experimental investigation," *Journal of Computer Information Systems* (41:2), 2001, pp. 99-109.
Pavlou, P., and Fygenson, M. "Understanding and predicting electronic commerce adoption: An extension of the theory of planned behavior," *MIS Quarterly* (30:1), 2006, pp. 115-143.
Rainer, R., Chimiak, W., Geisinger, K., Buss, D., Randall, L., and Wells, W. "A flexible telepathology system", in *Proceedings of IMAC 1995*, 1996, pp. 264-268.
Robey, D. "Identity, legitimacy and the dominant research paradigm: An alternative prescription for the IS discipline," *Journal of the Association for Information Systems,* (4:7), 2003, pp. 352-359.
Sanchez, D.M., Cavero, J.M., and Martinez, E.M. "The road toward Ontologies", In *Ontologies: A handbook of priciples, concepts and applications in information systems,"* Sharman, R., Kishore, R., and Ramesh, R. (Eds.), Springer, London, 2006, pp. 3-20.
Shanks, G., Tansley, E., and Weber, R. "Using ontology to validate conceptual models," *Communications of ACM* (46:10), 2003, pp. 85-89.
Spencer, D., and Hiltz, S. "A field study of use of synchronous chat in online courses," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003.
Schulzrinne, H. "World Wide Web: whence, whither, what next?" *IEEE Network* (10:2), 1996, pp. 10-17.
Thomas, B. "Recipe for E-commerce", *IEEE Internet Computing* (1:6), 1997, pp. 72-74.
Timpf, S. "Ontologies of wayfinding: A traveler's Perspective," *Networks and Spatial Economics* (2:1), 2002, pp. 9-33.
Turban, E., King, D., Viehland, D., and Lee, J. *Electronic Commerce 2006: A Managerial Perspective*. Pearson Prentice Hall, 2006
Orlikowski, W., and Iacono, C. "Research commentary: Desperately seeking the" IT" in IT research: A call to theorizing the IT artifact," *Information Systems Research* (12:2), 2001, pp. 121-134.

Osrael, J., Froihofer, L., and Goeschka, K. "Replication in service-oriented systems," *Software Engineering and Fault Tolerance. Pelliccione*, N. Guelfi, H. Muccini, A Romanovsky (Ed.) 2007.
Osterwalder, A., Pigneur, Y., and Tucci, C. "Clarifying business models: origins, present, and future of the concept," *Communications of AIS* (16), May 2005, pp. 1-25.
Osterwalder, A., and Pigneur, Y. "An e-Business model ontology for modeling e-Business," in *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.
Uschold, M., and King, M. "Towards a methodology for building ontologies," in *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.
Uschold, M., King, M., Moralee, S., and Zorgios, Y. "The enterprise ontology," *The Knowledge Engineering Review* (13:1), 1998, pp. 31-89.
VanderAalst, W. "Process-oriented architectures for electronic commerce and interorganizational workflow," *Information Systems* (24:8), 2000, pp. 639-671.
VanderMeer, D., Datta, A., Dutta, K., Thomas, H., Ramamritham, K., and Navathe, S. "FUSION: A system allowing dynamic Web service composition and automatic execution," in *Proceedings of the International Conference on E-Commerce, 2003,* pp. 399-404.
Venkatesh, V., Morris, M., Davis, G. and Davis, F. "User acceptance of information technology: Toward a unified view," *MIS Quarterly* (27:3), 2003, pp. 425-478.
Von Bertalanffy, L. *General System Theory: Foundations, Development, Applications*. George Braziller, 1968.
Wand, Y., and Weber, R. "An ontological model of an information system," *IEEE Transactions on Software Engineering* (16:11), 1990, pp. 1282-1292.
Wand, Y., and Weber, R. "On the ontological expressiveness of information systems analysis and design grammars," *Journal of Information Systems* (3:4), 1993, pp. 217-237.
Wand, Y., and Weber, R. "On the deep structure of information systems," *Information Systems Journal* (5:3), 1995, pp. 203-223.
Wand, Y., Storey, V., and Weber, R. "An ontological analysis of the relationship construct in conceptual modeling," *ACM Transactions on Database Systems* (24:4), 1999, pp. 494-528.
Wand, Y. ,and Weber, R. "Research commentary: Information systems and conceptual modeling: A research agenda," *Information Systems Research* (13:4), 2002, pp. 363-376.
Wand, Y., and Weber, R. "Reflection: Ontology in information systems," *Journal of Database Management* (15:2), 2004, pp. iii-vi.
Ward, J., Hemingway, C., and Daniel, E. "A framework for addressing the organizational issues of enterprise systems implementation," *Journal of Strategic Information Systems* (14:2), 2005, pp. 97-119.
Watson, H. J., Gerard, J. G., Gonzalez, L. E., Haywood, M. E., and Fenton, D. "Data warehousing failures: Case studies and findings," *Journal of Data Warehousing* (4:1), 1999, pp. 44-55.
Weber, R. *Ontological Foundations of Information Systems* (Vol. Monograph #4). Blackburn, Victoria: Buscombe Vicprint, 1997.
Yu, H., Spenko, M., and Dubowsky, S. "An adaptive shared control system for an intelligent mobility aid for the elderly," *Autonomous Robots* (15:1), 2003, pp. 53-66.