

Designing for Mutability in Information Systems Artifacts

Shirley Gregor

**School of Accounting and Business Information Systems,
The Australian National University
email: shirley.gregor@anu.edu.au**

Juhani Iivari

**Department of information Processing Science, University of Oulu
email: juhani.iivari@oulu.fi**

Abstract

This paper aims to extend understanding of the nature of information systems and technology (IS/IT) artifacts and the manner in which information systems design theories address the mutable nature of these artifacts. The term 'semizoa' is introduced to refer to IS/IT artifacts as mutable systems that exhibit some of the characteristics of living creatures and that are only in part designable. It is shown that the mutability of semizoa can be both constrained and enabled in IS design theories, using concepts of homeostasis, situated action, autopoiesis, learning, evolution, emergence and redesign. Consideration of the range and nature of these characteristics provides a source of ideas for systems designers in designing for specific IS meta-requirements. In addition, we show that IS design theories should include a reflective structural component dealing with the mutability of not only the system state but also its structure (schema). The paper contributes by addressing the lack of attention to the distinctive characteristics of IS/IT artifacts and by extending current ideas of design theories and theorising.

Introduction

Information systems (IS) is increasingly represented as a discipline that is concerned with the design, construction and use of artifacts based on information technology (IT) (see Weber, 1987; March and Smith, 1995; Dahlbom, 1996; Orlikowski and Iacono, 2001; Benbasat and Zmud, 2003; Hevner et al., 2004). The term 'artifact', however, tends to be used in a rather unreflective and undifferentiated manner and the distinctive characteristics of this class of artifact are not discussed. There is little critical examination of the assumptions that

underlie different treatments of artifacts. One assumption we believe should be questioned is the view that an IS/IT artifact is a relatively immutable result of a design process — an ‘end state’ that is arrived at as a result of a search process (see March and Smith, 1995; Hevner et al., 2004). A fixed end state is more characteristic of the inorganic artifacts that result from other design disciplines such as engineering (for example, a bridge, railway or painting).

Our argument is that IS/IT artifacts differ in fundamental ways from these other products of human design activity and that we need to talk about them and theorise about them in different ways. IS/IT artifacts are inherently *dynamic systems*. It was recognised early on that they evolve and change and that their real use may differ from their intended use (see Keen and Scott Morton, 1978). Dynamic systems are studied in many contexts and under many labels, including cybernetics, general systems theory, system dynamics, the systems approach and complex adaptive systems. Parallel lines of thought can be detected in structuration theory. Yet there is little if any attention paid in these fields to how such systems are *designed* — most effort is devoted into studying their behaviour as existing objects, following the natural science paradigm. The time is ripe to take ideas from the study of complex systems and see how they can be melded with ideas from the design science paradigm to give a much richer picture of design theorising in IS/IT. To date the borrowing from theories of complex systems has tended to be piecemeal and outside an encompassing framework.

The aim of this paper is to extend our meta-theoretical understanding of the nature of IS/IT artifacts as growing, changing and dynamic systems and the manner in which information systems design theories (ISDTs) can address the mutable nature of these artifacts. We provide a high-level framework for thinking about design theories that provide for the mutability in IS/IT artifacts in different ways. The need to explicitly include a component dealing with the mutability of designed artifacts in ISDTs is argued for in Gregor and Jones (2004, 2006).

We claim that a new way of thinking is required to capture design conceptualisations of IS/IT artifacts as complex phenomena that change and adapt in varying ways and to varying degrees. The term we introduce here to capture a more encompassing view of organic-type artifacts in IS and IT is *semizoa* a word derived from the Greek for living creatures (*zoa*). We define *semizoa* as IS or IT artifacts that exhibit the characteristic of mutability to some degree, that is, they grow, change (or are changed), and exhibit adaptive behaviour. Further, *semizoa* have the potential to modify, transform or constrain their surrounding environment.

Our work has theoretical significance because of the lack of prior attention to artifact mutability in the formal specification of design theories (as in Walls, et al., 1992, 2004). The paper also potentially has considerable practical significance.

We agree with van Aken (2004), who argues eloquently that one needs prescription-driven research that provides solutions for management problems in addition to description-driven research that enables us to understand the nature of problems, but leaves undone the task of developing interventions. Differing from van Aken, however, our focus lies in complex artifacts that require considerable design and development activity. A sounder and fuller basis for developing design theory for artifacts of this type offers opportunities for developing better theory to underpin design and development activities in practice.

Note that our investigation is meta-theoretical: it is not a specific theory but is a higher level analysis of a particular category of theorising — ‘design theorising’.

The composition of the paper is as follows. The following section reviews perspectives in IS on the IS/IT artifact and proposes a continuum between designed and natural artifacts and argues there is a need to recognise IS/IT artifacts as semiozoa. We then proceed by considering a number of the ways in which designing for mutability occurs with semiozoic artifacts. Our discussion leads to a number of suggestions as to how design theories and design theorising could benefit from our arguments. We conclude with some suggestions for further work.

The nature of IS artifacts

IS and IT artifacts

While it is currently quite common in the IS literature to talk about IT or IS artifacts, there is not a great deal of discussion of exactly what is meant by these terms and some divergence in views can be detected. Dahlbom suggested in 1996 that our focus should be on IT rather than IS artifacts because the latter do not easily cover, for instance, personal computing, communication, electronic publishing, air traffic control and intelligent houses. Subsequently, Orlikowski and Iacono (2001) popularised the phrase ‘IT artifact’ within the IS research community. These authors distinguished 13 different views of IT artifacts in the 188 articles published in the journal *IS Research* in the decade beginning in 1990 and ending in 1999. Most of these conceptualisations viewed IT artifacts as black boxes without looking inside the artifact. The articles reviewed focused on the building of IT artifacts with particular capabilities (the computational view of technology), their intended uses (the tool view of technology), technology as a variable (the proxy view of technology), and the interaction between people and technology (the ensemble view of technology). All IT artifacts were treated as a homogenous set in this review. As a whole, the discussions of IS/IT artifacts are characterised by somewhat convoluted definitions (see Orlikowski and Iacono (2001)) and Benbasat and Zmud (2003)) and some lack of recognition of IS as artifacts in themselves.

In this paper we will consider the range of artifacts that includes both IT and IS artifacts, the major focus lying in IT application artifacts. In the terms of Walls et al. (1992), March and Smith (1995) and Lee (1999), an information system is an instantiation of more general information technology. Information systems form a subset of IT artifacts, which obviously include various computer hardware and software artifacts. The word 'artifact' is used in the sense that it is an object designed, manufactured, used or modified by human activity.

Both IS and IT artifacts are also *systems*, where a system is:

Any portion of the material universe which we chose to separate in thought from the rest of the universe for the purpose of considering and discussing the various changes which may occur within it under various conditions (Gibbs, cited in the Principia Cybernetica Web, 2004).

Both an IS and an IT artifact qualify as a system because they have somewhere within their boundary a computer system that allows the artifact to change and exhibit mutability — the essential nature of a computer system being that it can be self-modifying. Information systems also exhibit mutability because they encompass the human users of technology, a further source of change. These basic definitional matters are important because we need to recognise that we have the interesting situation where the objects of interest for IS and IT are both artifacts *and* complex dynamic systems with the capability of self-modification.

From artifacts to semi-artifacts to semiozoa

Simon (1969/1996) makes a distinction between artificial or man-made things and natural things. He associates artifacts with design, in that they are designed (synthesised) by human beings, even though not necessarily with full forethought. We feel that the dichotomy between designed artifacts and natural objects is too simple. Many 'artifacts' are only partly the work of a designer.

Interestingly, Dahlbom (1996) adopts a very broad interpretation of the concept of artifact, claiming that 'People and their lives are themselves artifacts, constructed, and the major material in that construction is technology'. Referring more to IT, he continues:

When we say we study artifacts, it is not computers or computer systems we mean, but information technology use, conceived as a complex and changing combine of people and technology. To think of this combine as an artifact means to approach it with a design attitude, asking questions like: This could be different? What is wrong with it? How could it be improved?

The concept of artifact in this view implies that an artifact is at most partially man-made and designed. Perhaps it is more appropriate to say that they are cultivated rather than designed (Dahlbom, 2005).

Figure 1 gives an example of this continuum of ‘artifacts’ starting from completely designable artifacts such as mathematical theories and ending with natural objects in our environment that nevertheless are partly man-made because of factors such as cultivation, breeding, genetic engineering, and training.

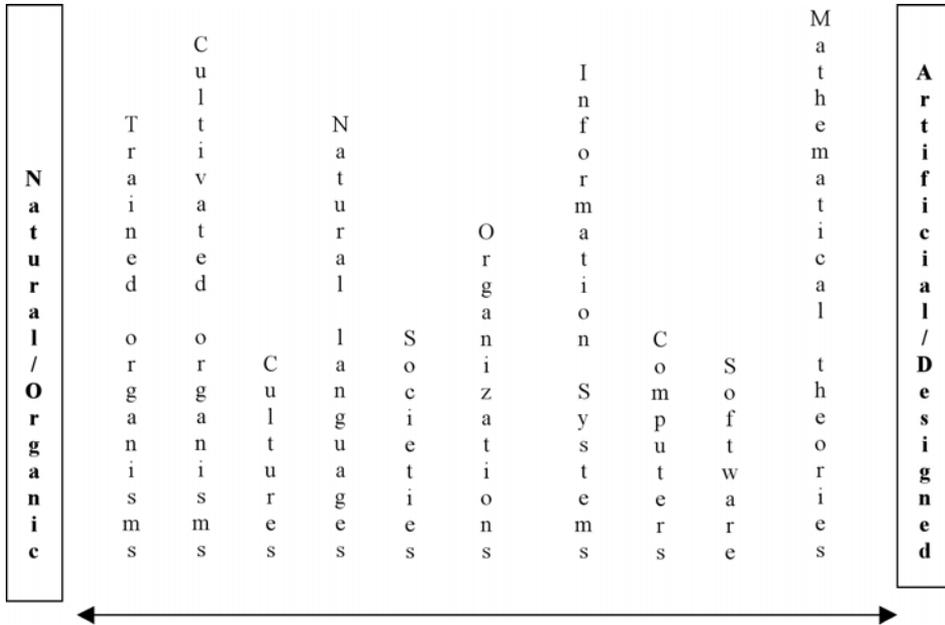


Figure 1: A natural-artificial continuum.

The positions of the different phenomena on the continuum of Figure 1 are only indicative. Software, in Figure 1, is interpreted as a system of algorithms, close to mathematical theories. Computers have a physical implementation that gives them a natural element. In addition to software and computers, IS comprise an information base that is only partially designable and makes IS additionally organic or emergent. On the opposite side, trained organisms (such as human beings and some animals) are considered less ‘designed’ because the influence of training is only ontogenetic with cultivated organisms, and the influence of cultivation and breeding is phylogenetic.¹ One should also observe that there may be internal variation within each phenomenon. For example, some natural languages such as Finnish may be more designed than others such as English. Similarly, some societies may be more designed than others and organisations can also differ in the degree to which they are designed.

Our purpose here is to illustrate that the traditional dichotomy between artificial and natural is a simplification. It is also obvious that the term ‘artifact’ emphasises

¹ Ontogenetic: Of, relating to, or characteristic of ontogenesis; relating to the development of the individual organism. Phylogenetic or phylogenetic: Relating to the race history of an organism or organisms (Oxford English Dictionary, Online version, 2004)

the artificial, designed end of the continuum but, unfortunately, we were unable to find a better existing term than 'artifact' that avoids this implication. 'Technology', when interpreted as 'a design for instrumental action that reduces the uncertainty in the cause-effect relationship involved in achieving a desired outcome' (Rogers, 1995), might be an alternative, but it has too technical a connotation. Järvinen (2004) prefers to speak about 'innovations' rather than 'artifacts', but the concept of an 'innovation' may lose the connotation of artificiality in contrast to natural or behavioural science theories. Recognising the bias in the concept of an artifact towards a static, designed object, we looked for an adjective to counterbalance this bias, ending with the adjective '*semizoic*'. The term we use for the class of IS/IT artifacts as a whole is *semizoa*, a term created from the Greek '*zoa*' for living creatures and '*semi*' for 'almost'. The singular is '*semizoan*' and the adjectival form is '*semizoic*'. We believe that this, perhaps paradoxical, phrase '*semizoic artifact*' better captures the richness of IS/IT artifacts.²

Figure 1 also suggests that information systems differ in their degree of artificiality from other IT artifacts such as software and computers. Many IT artifacts are only partly the work of a single designer. Systems are increasingly outcomes of distributed design where numerous designers engage in designing without being directly aware of each other. Many Web-based systems are examples of this. A resulting system may exhibit emergent features as an outcome of numerous local actions (for example, use, interpretation, negotiation and redesign), but these emergent features cannot be anticipated by reference to any *a priori* design. At a more theoretical level, the literature on the social construction of technology (Bijker et al., 1989; Bijker and Law, 1992; Orlikowski and Gash, 1994) discusses this emergent aspect of many artifacts. The provocative article of Truex et al. (1999) suggests that emergent organisations need continuous redevelopment of their systems but, in spite of the 'Growing systems in emergent organisations' title of their paper, the authors fail to recognise emergent information systems that grow without any continuous redevelopment. More recently, Markus et al. (2002) have analysed the provision of IT support for emergent knowledge processes (EKPs), which they define as organisational activity patterns characterised by:

- an emergent process of deliberations with no best structure or sequence;
- an actor set that is unpredictable in terms of job roles or prior knowledge; and
- knowledge requirements for general and specific distributed expertise.

² It is with some hesitation that we introduce this new term to a field with a proliferation of invented terms. Language, however, is influential. Our way of talking about things and the words we use can circumscribe thought. Particular words have particular connotations. It is preferable to recognise the special nature of IS/IT artifacts as '*semizoic artifacts*' by finding a new word for them.

To summarise the discussion to this point, we see IS/IT artifacts as complex systems that exhibit mutability. They are in part designable and in part they exhibit characteristics typical of organic life in that they change in ways that could not be completely anticipated. This special class of artifact we have given the label *semizoa*. In the next section we describe theories that inform the design of *semizoa* artifacts and some of the design mechanisms for achieving mutability.

Towards design theory for *semizoa* artifacts

Kernel theories

Our conceptualisation of IS/IT artifacts as complex systems means there is a wide array of reference disciplines to draw upon as ideas for design. Three reference disciplines are particularly promising sources of underlying kernel theories for the design of *semizoa* artifacts: biology, computer science and especially Artificial Intelligence (AI), and systems theory. The interactions among these fields is traced by Richardson (1991) who shows how a number of disciplines, including biology, engineering, mathematics and the social sciences came together in the Macy Cybernetics Conferences, with cybernetics subsequently leading to further streams of thought including control theory, information theory, digital computing, system dynamics and systems theory.

We have depicted *semizoa* as only in part designable and also as exhibiting growth and change in ways that parallel the mutability of organic life forms. Thus, it is not surprising biology forms a significant source of kernel theories for *semizoa*. Biology has provided a number of ideas and metaphors for computing. De Castro and Von Zuben (2004) suggest that ‘natural computing’ encompass these three types of approaches:

- biologically-inspired computing;
- artificial life and fractal geometry of nature; and
- computing with natural means.

Similar ideas are expressed by Christopher Langton (1989), the ‘father’ of artificial life (*Alife*), who describes the concept as ‘the study of man-made systems that exhibit behaviours characteristic of natural living systems’.

Computer scientists, especially in the field of AI, have adopted ideas and metaphors for computing from biology to develop computational systems and algorithms capable of solving complex problems; for example, with artificial neural networks and evolutionary algorithms. More fundamentally, the increased distribution of computing has led computer scientists to develop new paradigms for computing based on interaction rather than algorithms (Wegner, 1997; Wegner and Eberbach, 2004). The interaction paradigm, in which one sees computation as a distributed ongoing interaction of interaction machines, provides a theoretical model of computation that is more consistent with the

idea of mutability, change and evolution than the algorithm paradigm based on the Turing machine.

Even though biologically inspired computing, artificial life and artificial intelligence provide a bundle of technologies, many of them reflect the algorithmic paradigm of search, problem solving and optimisation, rather than an interaction paradigm. To analyse and understand the mutability and change of semizoic artifacts, systems theory provides a third perspective (von von Bertalanffy, 1973; Ashby 1956). IT artifacts are systems and, as digital systems, very complex ones (Parnas, 1985). Systems theory has paid considerable attention to the analysis of complex and dynamic systems (see Aulin, 1989; Bar-Yam, 1997).

These potential sources of kernel theories for ISDT provide for many concepts that can be recognised in existing designs for semizoic artifacts. Characteristics of these artifacts include homeostatic mechanisms, situated action, autopoiesis, learning, evolution and emergence; although not necessarily all of them together, but at least one of them. These requirements constrain or enable mutability in the semi-designed artifact, as explained further in the following section.

Designing for mutability in semizoic artifacts

In this section we examine a number of the ways in which mutability can be enabled or constrained in semizoic IS/IT artifacts. These different categories of mutability have been identified by studying the theories discussed above and the different ways in which the mutability of semizoic artifacts are dealt with in existing design theories. We do not claim that our list is exhaustive, although we believe that it captures the most salient aspects of providing for mutability. Nor do we claim that the different categories we present are mutually exclusive; some IS/IT artifacts will have several types of mutability allowed for. In keeping with our previous ideas about designed and semi-designed artifacts, some types of mutability can be more 'designed-in' than others. With some types the designer can set up some initial pre-conditions that allow for change, but it is unlikely to be possible to anticipate completely the direction change will take.

Nilpotence

A nilpotent system is a memory-less system (without any persistent information base) that gives the same response to the same stimulus and returns to its initial rest-state after a finite number of units of time (Aulin 1989; Järvinen 2004). Many real-time computer systems are nilpotent systems. They are designed to react to different external events, to give the same response to the same type of event, and to return to the idle state after processing the external event. We are including nilpotent systems to demonstrate a dimension of minimum mutability in a designed artifact.

Homeostatic mechanisms

Biologists use the term homeostasis for the state where an organism is relatively insensitive to its environment. That is, it has mechanisms that allow it to function despite wide variability in external conditions. Homeostasis normally refers to the maintenance of the internal environment of the body within narrow and rigidly controlled limits (for example, a body temperature of around 37°C in human beings).

Computers are, of course widely used as control systems where the goal is to maintain the homeostatic state of the controlled process, such as keeping the temperature of a process within specified limits. Information systems can be designed to exhibit homeostatic behaviour. For example, the aim might be to control the response time of a distributed database. If the response time approaches an allowed upper limit, the system may automatically reorganise its database. In this way mutability is restricted.

Autopoiesis

Living beings are characterised by their autopoietic organisation. The term autopoiesis was coined in cybernetics by Humberto Maturana to refer to a special case of homeostasis in which the critical variable of the system that is held constant is that system's own organisation (Maturana and Varela, 1980). Maturana recognised that the autonomous quality of the living cell or organism was captured by the term self-creation, self-making, or self-producing. The organism is capable of maintaining its own organisation as a unity, in terms of its components and the relationships between them, without outside interference. That is, the basic structure of the organism remains the same. In simple terms, all the cells in a living creature might change, but it appears to have much the same appearance to an observer. An important aspect of autopoietic theory is that there is a boundary around the system. It can obtain matter and energy from its environment while at the same time being operationally closed. Closed, that is, to instruction or control. Further explanation of this concept and its relevance to IS and IT can be found in Mingers (1989, 1994) and Winograd and Flores (1986).

Application of the concept to IS design is not straightforward. In terms of the theory of Maturana and Varela, the focus should be on the internal dynamics of the system, and not on an ascription of outwardly-focused behaviour such as recognition and reaction to external events. An example with living organisms is the immune system's distinction between self and non-self. An analogy with a database system would be the principles of self-organisation that it is given to maintain database integrity. For example, it might have a consistency rule that says a department has only one manager. The system should then never contain data that violates this consistency rule.

Situated action

The idea of ‘situated action’ has been discussed in a number of contexts — in biological systems, but also in robotics, human-computer interaction design and with parallels in structuration theory and actor-network theory (see Johnston, 2001). We will mainly base our discussion on the work of Lucy Suchman (1987), who introduced the idea in the context of design theory for human-computer interaction. This design theory sees people’s behaviour as contextualised; that is, the situation in part determines what they will do. In situated action individuals continuously monitor their flow of action and the context in which the action takes place.

To what extent can IS/IT artifacts exhibit situated action in the sense of Suchman (1987)? Lieberman and Selker (2000) discuss context-aware systems that adapt to and learn from context.³ Erickson (2002) expresses serious concerns about context-aware artificial systems, about their capability to sense their rich context in a way comparable to human beings, and about their capability to intelligently select an action appropriate in the context. And when one takes into account that situated action as introduced by Suchman (1987) comprises artful improvisation (Orlikowski and Hofman, 1997) as an essential aspect, the possibilities of artificial situated action become still slimmer.

Despite these difficulties, some researchers have applied the concepts of situated action to the design of information systems, showing how, in an environment such as a manufacturing plant, the need for high-level planning of interrelated complex processes can be reduced by allowing some of the actors in the system to respond in a situated way to their environment (Johnston, 1995, 2001; Johnston and Milton, 2002). One of the authors, in the course of writing this paper, realised that a situated action perspective could help in another project involving designing ‘zero intelligence’ trading agents and decision aids in a complex trading environment.

Learning

A further characteristic that distinguishes intelligent life is the capacity to learn and acquire knowledge. The attempt to model this characteristic is one that occupies designers who develop theories of machine learning and build ‘systems that learn’ (see Jain et al., 1999). Machine learning has, of course, a long tradition in AI (Langley and Simon, 1995) but it is beyond the scope of the present paper to review this research tradition.

All information systems that have a memory (information base) have a potential for learning in the simple sense that they can accumulate more information into their information base. A more advanced form of learning implies changes in

³ In our view context-awareness does not necessarily include learning even though it often does.

the structure and functionality of the system at the IS schema level. For example, the interconnections of system components and the behaviour of components may be changed because of learning (as in, for example, neural computing).

An example of an IS incorporating learning is provided by Hall et al (2003) who propose a theory for the design of learning-oriented knowledge management systems.

Evolution

The term 'evolution' is used with a number of meanings in information systems and software engineering, often without any connection to biological evolution (for examples, see Hawgood, 1982; Lehman and Belady, 1985). Generally, 'evolution' refers to a trial-and-error process of variation and natural selection of systems at all levels of complexity. In the Darwinian theory of biological evolution, 'natural selection' is distinguished from 'artificial' selection where, for example, the breeding of animals is controlled so that certain traits are retained or eliminated.

Swanson (1982) discusses to what extent biological concepts such as genotypes and phenotypes and natural selection apply as analogies to information systems. He parallels design with genotype, implementation with phenotype, and utilisation with natural selection.⁴ In the case of design (genotype) he points out that, contrary to biological organisms, which have a fixed genotype, information systems may be redesigned during their implementation, and concludes that effective IS design must provide for its own conceptual adaptation. In the context of implementation (phenotypes) he mainly discusses modifications during implementation, arguing that, instead of a 'faithful' implementation, 'a strong case can be made for a "faithless" implementation which corrects in action that which has been misconceived in thought'. Finally, in the context of utilisation Swanson (1982) notes that over time utilisation tends to decline, not necessarily because of any observable failings in system implementation, but because the user may be better served by a new IS design (genotype).⁵ Helylighen (1997a, 1997b) provides a further interpretation of evolution in which there is no need for competition between simultaneously present configurations.

Evolutionary concepts underlie the design architectures of some computer systems. Evolutionary and genetic algorithms (Chaudry et al., 2000) model the

⁴ Genotype: The genetic constitution of an individual, esp. as distinguished from its phenotype; the sum-total of the genes in an individual or group. Phenotype: A type of organism distinguishable from others by observable features; the sum total of the observable features of an individual, regarded as the consequence of the interaction of its genotype with its environment (Oxford English Dictionary, 2004)

⁵ Swanson's discussion is weakened by the fact that he does not make a difference between the IS model (schema) and the IS state. Applying this distinction, one can interpret a genotype as an IS model, even though an information system may experience a number of redesigns during its life, and a phenotype as an IS state.

evolution of a population of individuals and provide methods for solving optimisation problems. Evolutionary concepts have also been utilised in the context of software architectures, to model how software systems can evolve (e.g. Paderewski-Rodríguez et al., 2004) and Scharl (2000) describes evolutionary methods for Web development.

Emergence

The terms ‘emergence’ and ‘emergent’ are finding increasing use in the field of information systems, although the labels are on occasion applied rather loosely. In systems theory the concepts encapsulated are those of the ‘whole being more than the sum of its parts’ or the ‘generation of complex systems from simple ones’ (see Bar-Yam, 1997, for a more detailed analysis).

John Holland (1996, 1999), the originator of genetic algorithms, proposed that the study of emergence was relevant to the development of complex systems in the arts, business, the evolution of society and the generation of new ideas. Problems studied include the evolution of organisations — how a collection of initially independent but interacting agents can come to organise themselves so as to form a coordinated system that can tackle problems too complex for the individual agents themselves (Heylighen 1997b).

One can identify emergence at different levels — at the structural level or in the changes of state that arise. Knowledge Management Support Systems such as *Answer Garden* (Ackerman, 1998) provide good examples of emergent information systems in which the support provided by the information system is much more dependent on the growth of the system than on its design. A system like *Answer Garden* is a learning system in the sense that it can accumulate more knowledge into its knowledge base but it may also exhibit emergence. To illustrate, existing knowledge in the knowledge base may be like a puzzle where pieces of knowledge are interrelated, but do not quite fit together. However, a new piece of knowledge may be the critical missing piece in the puzzle that allows the integration of the existing pieces of knowledge and their interpretation in a new light. This case shows emergence at the level of the IS state.

Information systems in which the design is distributed over a number of decisions illustrate emergence at the IS structural level. Imagine a hypertext or multimedia information system on certain topics to which several people can provide content. Their design decisions to insert hyperlinks form an emergent structure that cannot be predicted in advance. The World-Wide Web, with its distributed hypermedia architecture, is an example of an emergent system on a large scale.

In IS/IT systems, emergence cannot be totally planned or designed in. Yet, conditions can be set up that allow emergence to develop (like the plan for a garden). Examples are the standards for interfaces that allow new systems to join to existing systems, and open systems.

Redesign as a response to externally initiated change

Unlike the previous examples of change in an artifact, and reflecting their 'designed' rather than 'natural' creation, some changes to IS/IT artifacts do not have strong parallels with those that occur in living creatures.

After its original design and implementation, an IS/IT artifact will generally be subjected to further change and modification to meet changing requirements, to correct errors and to allow re-design by new designers. The ability to change in this respect is mostly a desirable aspect of computer software and systems. That is, they should be easy to modify and maintain, and often substantial modification is required. Examples of similar occurrences with living systems are not common. Perhaps the closest is genetic manipulation, or an artificial limb, or a bionic ear.

A number of influential design theories for programming and systems construction were motivated by the need to make programs easy to maintain, modify and change. Many of these design theories make use of the idea that change can be more easily accomplished if it is limited to one section or module of a program. Thus, we have the ideas of modularisation and module de-coupling in structured program design, and encapsulation in object-oriented methods.

A further example is the database concept of 'data independence', which involves the idea that the internal schema of a database can be changed without any change implications for the conceptual schema. Also, Codd's provision of views in relational database design (Codd, 1970, 1982) means that the database can appear differently to different users at different times and those users can adapt the views from outside as they wish. Further, in the case of relational databases, the database schema can be changed while use is ongoing.⁶

Implications for IS design theorising

The review of the different mechanisms for providing for the mutability of semioic artifacts leads us to some conclusions as to how design theories for ISDT should be specified. Walls et al. (1992) omitted Dubin's (1978) concept of 'system states' in their formulation of ISDT components. In Dubin's terms, one should specify in a theory what states of a system will be covered by a theory that is proposed (see Gregor and Jones, 2004, 2006, for further elaboration). Our discussion above demonstrates that such a component is valuable in ISDT because they deal with changing semioic artifacts, which will almost certainly demonstrate some form of mutability over their life and thus changes in state. We argue that an ISDT will be improved if the proponents of a theory consciously

⁶ Note, however, that even though it is easy to redesign the relational schema, it is not that easy to get the system state to correspond to the new schema. This requires that existing tuples be extended to include attribute values of the new attributes. Relational DBMSs manage this by inserting null values into the new attributes.

reflect on the degree of change they anticipate for their designed artifacts. Some degree of change may be provided for deliberately in the meta-requirements (as in being flexible in the face of future amendments). In addition, however, the theorist should document what changes in state the theory will cover.

A further interesting conclusion can be drawn by careful study of the nature of the changes that can occur in achieving the meta-requirements listed in the previous section. It is not only system states that can change but also the basic structure of the system itself. One way of conceptualising these broad directions in which information systems change is to think of an IS schema or model (structure and functions of the system) in addition to the IS states that the system can occupy at different times.⁷ When thinking of the way in which an information system changes, we can think of changes both to:

- its model/schema (its basic form and functional capacities), and
- its state (i.e. the changes as it moves from one state to another over time).

A system's model/schema (its basic form) is related to its design and is the subject of design theory in IS, but it is also significant to recognise state changes. A system's capability to change its structure (IS schema) requires that the system has a reflective capability, including a self-representation (Maes, 1987). The system has a model of itself (schema) and it has the capacity to change itself by its computation (Maes, 1987).

Table I summarises our findings by showing the types of mutability discussed and their appearance at the levels of the IS schema and IS state. With some types of mutability the changes to the schema (structure) can be anticipated to some extent and facilitated (as with re-design), or the degree of change can be limited (as with nilpotent systems, homeostasis and autopoiesis). With other types of mutability the extent or nature of change cannot easily be anticipated, although the designer may set up conditions for change to take place (learning, evolution and emergence), meaning that the resultant system is only in part designed.

Concluding remarks

We have, in this paper, highlighted the varying degrees to which IS/IT artifacts can be 'designed' and how these artifacts can be viewed as occupying a space along a continuum that exists between artificial, completely designed artifacts, and more natural, organic entities, which may yet also be partly designable.

⁷ More formally, the IS model can be conceived as the pair <Information base schema, Process schema>. The IS state $s \in R \times D$ (Information base schema) $\times D$ (Process schema). D (Information base schema) describes the potential lawful states of the information base and D (Process schema) the possible execution states of the software (programs) in the system.

Type of mutability	Nature of change	
	Changes in the IS structure/schema	Changes in the IS state
Nilpotent systems	No change.	Returns to the single rest (idle) state.
Homeostasis	No change.	Maintains the state within specified limits.
Autopoiesis	Limited change.	Maintains its own organisational state.
Situated action	Possible change in IS structure/schema (e.g. new learned response).	Changes state to correspond to the context (situation) and responds depending on the context.
Learning	Structural and functional change, with gaining of new knowledge.	New states by accumulating new knowledge in the knowledge base.
Evolution	Competitive selection among new designs.	Competition between potential successive states.
Emergence	Emergent structure and functionality because of parallel, interdependent design decisions that shape the IS schema without any <i>a priori</i> design.	Emergent states as new data, knowledge and/or activity occur.
Redesign	Structural and functional change in the IS model, as determined by the external designer.	Dependent on changes to the schema

Table 1: The mutability of IS/IT artifacts and the nature of change

We introduced the term *semizoa* to refer to these organic-type IS/IT artifacts, which exhibit the property of mutability to some degree; that is they grow, change or are changed, and exhibit adaptive behaviour to some degree. The properties of mutability that IS/IT artifacts possess are identified as being a consequence of the essence of their nature, which is that of computer systems and living beings. Varying types of mutability have been explored, drawing on work in a number of disparate areas, including systems theory, systems dynamics, complexity theory, sociology, and IS/IT design theories. The mechanisms for providing for mutability examined included nilpotence, homeostasis, situated action, autopoiesis, learning, evolution, emergence and re-design. Our exploration shows that many metaphors used in connection with IS/IT artifacts are borrowed from the study of living creatures. The dimensions of mutability are not mutually exclusive, and a single *semizoa* artifact could exhibit a number of these characteristics.

Implications of the paper

Our paper has significance in that, despite recent attention having been paid to the artifactual nature of IS and IT, the term ‘artifact’ has been used with little reflection and without differentiating among various types of artifacts. That is, there has been relatively little attention paid to the ontology of IS/IT artifacts. Our paper provides a unifying perspective on ideas that have originated in quite different paradigms, but can be recognised in different forms across many fields. The perspective that brings all these ideas together is the recognition that they all deal with different aspects of change, or mutability, which is a distinguishing characteristic of systems, living creatures and *semizoa* IS/IT artifacts, and all have parallels in ideas for design of IS/IT.

The paper has further significance in the implications for design theorising for semizoic artifacts. Our analysis suggests that:

- An IS artifact should not be regarded as a static goal that is the end product of a search process. Our many examples show the mutability of these artifacts. Designers should consider establishing the basis for a design trajectory, rather than aiming at a completely designed finished artifact. The partially-designed nature of some semizoa, for example those that exhibit emergence, means that the final form and behaviour of the artifact can not be specified in advance. The challenge is to find design principles that allow 'desirable' forms and behaviours to develop.
- Some of the characteristics of semizoa can be explicitly specified as design meta-requirements: for example, the ability to acquire new information (learning) or the requirement for systems to be easily modifiable, or extensible. The discussion in this paper shows some of the kernel theories that can inform designing for these requirements.
- Other characteristics of semizoa, while they may not be explicitly thought of as first-order, primary design goals, can help a semizoic artifact accomplish a first-order goal. For example, in the case of a trading agent in a stock market, the main goal is to maximise gains from trading. Considering how living organisms cope in situations of extreme complexity suggests that situated action concepts can help design an effective agent (AI researchers have been using these same concepts successfully in robotics).
- Designers should note that some of the dimensions of mutability may be mutually antagonistic. For example, the maintenance of self-stabilisation, or an autopoietic condition, is opposed to a condition where a semizoan is open to externally-originated modification or change, which could destroy or irrevocably alter its essential nature.
- IS designers and IS design theories should explicitly reflect on the dimensions of mutability that have been considered when they specify their theories, both for system structure and for system states. In this way what the theory encompasses is specified more fully, and stronger and more practical theory should ultimately result.
- The range in forms of mutability provides a fertile source of ideas, a 'menu', for new designs and new design theories.

Many of the ideas in this paper are not new. A number of existing approaches recognise that information systems are dynamic and can exhibit emergent behaviour (eg. Orlikowski and Gash, 1994). This prior work, however, tends to stop at the stage of analysing and describing some aspects of the behaviour of these systems: it does not go on to show how designers can explicitly confront and manage mutability through a variety of means, as we have tried to do here.

Further research: semiozoa as actors

The above analysis is the first attempt to analyse information systems and information technology as semiozoic artifacts. Obviously it can be deepened in many details. One aspect, not addressed in this paper is that semiozoic IS/IT artifacts have the potential to modify, transform, or constrain their surrounding environment. They can also be thought of as actors that cause or bring about change, and their success may depend on the extent to which they are able to serve in this role. To illustrate, the success of a Knowledge Repository System for a certain domain (topic) may depend on to the extent to which the system is able to stimulate the domain experts to contribute to the knowledge repository. If an expert finds the existing knowledge in the domain and the debate related to its validity is stimulating and rewarding, he or she may be more motivated to contribute (as with scientific debate and progress). Relevant here is actor-network theory, which is a high-level meta-theory that views non-human entities as actors (Law, 1992). Another high-level meta-theory that deals with the interaction between actions, whether or not of human origin, and structure (environment) is structuration theory (Giddens, 1984).

Further, Heidegger (1993) was concerned that modern technology, as opposed to previous technologies, modifies and challenges the natural order. It controls and reorders the natural order rather than simply using it. Heidegger uses the comparison between a windmill, which harnesses the wind but doesn't change it, and a hydro-electric dam, which captures and changes the river. The objects within the natural order are modified to become a standing reserve for technology. The water of the river becomes the power source for hydro-electric power generation. The way in which these objects are perceived is framed with a different perspective provided by modern technology. Heidegger saw the 'enframing' (*ge-stell*) offered by modern technology and its capacity to overwhelm and restrict all other ways of revealing as the essence and danger of modern technology. Heidegger's concerns were expressed before computer technology was much in evidence but they could be expected to hold with even more force given the increasing pervasiveness of IS/IT artifacts. Heidegger offers some possibility of a counterbalance to the pessimistic outlook of enframing through *poiesis*. However, different meanings are given to this concept in Heidegger's own writing and it is almost impossible to represent it clearly, especially in a limited space as here. Nevertheless, a simplistic attempt to do so might run along the lines that humankind can perhaps escape from the technical order through thinking and poetry (*poiesis*), which provide a different manner of revealing from that of technology.

A further challenge is to consider how we design for emergence. The coupling of ideas between what can be observed with existing emergent systems such as

the Web, and ideas from the field of complex and dynamic systems may be worthwhile.

To conclude, we present the idea that regarding IS/IT artifacts as semiozoa and dealing with their forms of mutability provides real prospects for the development of 'grand theories' of information systems that we would like to achieve. Some existing design theories that consider aspects of mutability have been influential, including Codd's relational database design theory, structured systems analysis and design theories, concepts of situated action, and object-oriented approaches. However, unlike prior work we have brought together a range of different forms of mutability, showing how they differ and how they can all be used in design work. We believe that building design theories that explicitly deal with the mutability of designed semiozoic artifacts provides a means of differentiating IS and allied fields from other design disciplines and can give our discipline a more readily identifiable theoretical base.

References

- Ackerman, M. 1998, 'Augmenting organisational memory: A field study on Answer Garden', *ACM Transactions on Information Systems*, vol. 16, no. 3, pp. 203-24.
- Ashby, W. R. 1956, *An Introduction to Cybernetics*, Chapman and Hall, London.
- Aulin, A. 1989, *Foundations of mathematic system dynamics: The fundamental theory of causal recursion and its application to social science and economics*, Pergamon Press, Oxford.
- Benbasat, I. and Zmud, R. W. 2003, 'The identity crisis within the discipline: Defining and communicating the discipline's core properties', *MIS Quarterly*, vol. 27, no. 2, pp. 183-94.
- Bar-Yam, Y. 1997, *Dynamics of Complex Systems*, Addison-Wesley, Reading, MA.
- Bijker, W. E., Hughes, T. P. and Pinch, T. J. (eds) 1989, *The Social Construction of Technological Systems, New Directions in the Sociology and History of Technology*, The MIT Press, Cambridge, MA.
- Bijker, W. E. and Law, J. (eds) 1992, *Shaping Technology/Building Society*, The MIT Press, Cambridge, MA.
- Chaudry, S. S., Varano, M. W. and Xu, L. 2000, 'Systems research, genetic algorithms and information systems', *Systems Research and Behavioral Science*, vol. 17, no. 2, pp. 149-62
- Codd, E. F. 1970, 'A relational model of data for large shared data banks', *Communications of the ACM*, vol. 13, no. 6, pp. 377-87.

- Codd, E. F. 1982, 'Relational database: A practical foundation for productivity. The 1981 Turing Award Lecture', *Communications of the ACM*, vol. 25, no. 2, pp. 109-17.
- De Castro, L. N. and Von Zuben, F. J. 2004, 'From biologically inspired computing to natural computing', in De Castro, L. N. and Von Zuben, F. J. (eds), *Recent Developments in Biologically Inspired Computing*, Idea Group Publishing, Hershey, PA.
- Dahlbom, B. 1996, 'The new informatics', *Scandinavian Journal of Information Systems*, vol. 8, no. 2, pp. 29-48.
- Dahlbom, B. 2005, Personal communication, June 13.
- Dubin, R. 1978, *Theory Building*, Revised ed., Free Press, London.
- Erickson, T. 2002, 'Some problems with the notion of context-aware computing, ask not for whom the cell phone tolls', *Communications of the ACM*, vol. 45, no. 2, pp. 102-4
- Giddens, A. 1984, *The Constitution of Society: Introduction of the Theory of Structuration*, University of California Press
- Gregor S. and Jones, D. 2004, 'The formulation of design theories', in Linger, H., Fisher, J., Wojtkowski, W., Zupancic, J., Vigo, K. and Arold, J. (eds) *Constructing the infrastructure for the knowledge economy: Methods and tools, theory and practice*, New York, Kluwer Academic.
- Gregor, S. and Jones, D. 2006, 'Improving the specification of Information Systems design theories', Working Paper, The Australian National University.
- Hall, D., Paradice, D. and Courtney, J. 2003, 'Building a theoretical foundation for a learning-oriented management system', *Journal of Information Technology Theory and Application*, vol. 5, no. 2, pp. 63-85.
- Hawgood, J. (ed.) 1982, *Evolutionary Information Systems*, North-Holland.
- Heidegger, M. 1993, 'The question concerning technology', in *Basic Writings*, Harper, San Francisco, pp. 311-41, translated from Martin Heidegger 1954, *Vortrage and Aufsätze*, Gunther Neske Verlag, Pfullingen, pp 13-44.
- Helylighen, F. 1997a, 'Basic concepts of the systems approach', in Heylighen, F., Joslyn, C. and Turchin, V. (eds), *Principia Cybernetica*, Brussels, viewed 1 November, 2004, <<http://pespmc1.vub.ac.be/SYSAPPR.html>>.
- Helylighen, F. 1997b, 'Evolutionary theory', in Heylighen, F., Joslyn, C. and Turchin, V. (eds), *Principia Cybernetica*, Brussels, viewed 1 November, 2004, <<http://pespmc1.vub.ac.be/EVOLUT.html>>.

- Hevner, A., March, S., Park, J. and Ram, S. 2004, 'Design science in information systems research', *MIS Quarterly*, vol. 28, no. 1, pp. 75-105.
- Holland, J. H. 1996, *How Adaptation Builds Complexity*, Addison-Wesley.
- Holland, J. H. 1999, *Emergence: From Chaos to Order*, Addison-Wesley.
- Jain, S., Osherson, D., Royer, J., and Sharma, A. 1999, *Systems That Learn* (2nd ed.), Bradford Books.
- Järvinen, P. 2004, *On Research Methods*, Opinpajan kirja, Tampere, Finland.
- Johnston, R. B. 1995, 'Making manufacturing practices tacit: A case study of computer aided production management and lean production', *Journal of the Operational Research Society*, vol. 46, no. 10, pp. 1174-83.
- Johnston, R. B. 2001, 'Situated action, stucturation and actor-network theory: An integrative perspective', in *Proceedings of ECIS 2001 the 9th European Conference on Information Systems*, Bled, Slovenia.
- Johnston, R. B. and Milton, S. K. 2002, 'The foundational role for theories of agency in understanding of information systems design', *Australian Journal of Information Systems*, vol. 9, Special Issue, pp. 40-9.
- Keen. P. and Scott Morton, M. 1978, *Decision Support Systems: An Organisational Perspective*, Reading, Addison-Wesley.
- Langley, P. and Simon, H. A. 1995, 'Applications of machine learning and rule induction', *Communications of the ACM*, vol. 38, no. 11, pp. 55-64.
- Langton, C. G. 1989, 'Artificial Life', in Langton, C. G. (ed.), *Artificial Life*, Volume VI of SFI Studies in the Sciences of Complexity, Addison-Wesley, Redwood City.
- Law, J. 1992, 'Notes on the theory of the actor-network ordering: Strategy and homogeneity', *Systems Practice*, vol. 5, no. 4, pp. 379-93.
- Lee, A. S. 1999, 'Researching MIS', in Currie, W. L. and Galliers, R. (eds) *Rethinking Management Information Systems*, Oxford University Press.
- Lehman, M. M. and Belady, L. A. 1985, *Program Evolution: Processes of Software Change*, Academic Press.
- Lieberman, H. and Selker, T. 2000, 'Out of context: computer systems that adapt to, and learn from, context', *IBM Systems Journal*, vol. 39, nos. 3 and 4, pp. 617-32.
- Maes, P. 1987, 'Concepts and experiments in computational reflection', *OOPSLA'87 Proceedings*, pp. 147-55.
- March, S. T., and Smith, G. F. 1995, 'Design and natural science research on information technology', *Decision Support Systems*, vol. 15, pp. 251-66.

- Markus, M., Majchrzak, L. A., and Gasser, L. 2002, 'A design theory for systems that support emergent knowledge processes', *MIS Quarterly*, vol. 26, pp. 179-212.
- Maturana, H. R. and Varela, F. J. 1980, *Autopoiesis and Cognition*, D. Reidel, Dordrecht, Holland.
- Mingers, J. 1989, 'An introduction to autopoiesis — implications and applications', *Systems Practice*, vol. 2, no. 2, pp. 159-80.
- Mingers, J. 1994, *self-producing systems: implications and applications of autopoiesis*, Plenum Publishing, New York.
- Orlikowski, W. J. and Gash, D. C. 1994, 'Technological frames: Making sense of information technology in organisations', *ACM Transactions on Information Systems*, vol. 12, no. 2, pp. 174-207.
- Orlikowski, W. J. and Hofman, J. D. 1997, 'An improvisational model for change management: The case of groupware technologies', *Sloan Management Review*, vol. 38, no. 2, pp. 11-21.
- Orlikowski, W. J. and Iacono, C. S. 2001, 'Research commentary: Desperately seeking the "IT" in IT research — A call to theorising the IT artifact', *Information Systems Research*, vol. 12, no. 2, pp. 121-34.
- Paderewski-Rodríguez, P., Torres-Carbonell, J. J., Rodríguez-Fortiz, M. J., Medina-Medina, N. and Molina-Ortiz, F. 2004, 'A software system evolutionary and adaptive framework: application to agent-based systems', *Journal of Systems Architecture*, vol. 50, pp. 407-501.
- Parnas, D. L. 1985, 'Software aspects of strategic defense systems', *Communications of the ACM*, vol. 28, no. 12, pp. 1326-35.
- Principia Cybernetica Web 2004, viewed November, 2004 <<http://pespmc1.vub.ac.be/DEFAULT.html>>.
- Richardson, G. P. 1991, *Feedback thought in social sciences and systems theory*, University of Pennsylvania Press, Philadelphia.
- Rogers, E. M. 1995, *Diffusion of innovations*, (4th ed.), The Free Press, New York.
- Scharl, A. 2000, *Evolutionary Web development*, Springer.
- Simon, H. 1996, *The sciences of the artificial*. (3rd ed.), MIT Press, Cambridge, MA, (1st ed., 1969, 2nd ed., 1981).
- Suchman, L. A. 1987, *Plans and situated actions: the problem of human-machine communication*, Cambridge: Cambridge Press.
- Swanson, E. B. 1982, 'A view of information system evolution', in Hawgood, J. (ed.), *Evolutionary information systems*, North-Holland.

- Truex, D., Baskeville, R., and Klein, H. 1999, 'Growing systems in emergent organisations', *Communications of the ACM*, vol. 42, no. 8, pp. 117-23.
- Van Aken, J. 2004, 'Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules', *Journal of Management Studies*, vol. 41, no. 2, pp. 219-46.
- von Bertalanffy, L. 1973, *General system theory* (revised ed.), George Braziller, New York.
- Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. 1992, 'Building an information system design theory for vigilant EIS', *Information Systems Research*, vol. 3, no. 1, pp. 36-59.
- Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. 2004, 'Assessing information system design theory in perspective: How useful was our 1992 rendition?', *Journal of Information Technology Theory and Practice*, vol. 6, no. 2, pp. 43-58.
- Weber, R. 1987, 'Toward a theory of artifacts: a paradigmatic base for information systems research', *Journal of Information Systems*, vol. 1, pp. 3-19.
- Wegner, P. 1997, 'Why interaction is more powerful than algorithms', *Communications of the ACM*, vol. 40, no. 5, pp. 80-91.
- Wegner, P. and Eberbach, E. 2004, 'New models of computing', *The Computer Journal* vol. 47, no. 1, pp. 4-9.
- Winograd, T. and Flores, F. 1986, *Understanding computers and cognition*, Addison-Wesley, Reading, MA.