Communications of the Association for Information Systems

Volume 28 Article 24

5-2011

ECIS 2010 Panel Report: Humanities-Enriched Information Systems

Jan H. Kroeze
School of Computing, University of South Africa, Pretoria, South Africa, jan.kroeze@gmail.com

Hugo H. Lotriet
School of Computing, University of South Africa, Pretoria, South Africa

Nehemiah Mavetera
School of Economics and Decision Sciences, North-West University (Mafikeng Campus), South Africa

Mark S. Pfaff
School of Informatics, Indiana University (Indianapolis Campus), Indianapolis, IN, US

Dirk J.v.R. Postma
Department of Education Studies, University of Johannesburg, Johannesburg, South Africa

See next page for additional authors

ECIS 2010 Panel Report: Humanities-Enriched Information Systems

Authors
Jan H. Kroeze, Hugo H. Lotriet, Nehemiah Mavetera, Mark S. Pfaff, Dirk J.v.R. Postma, Kosheek Sewchurran, and Heikki Topi

Communications of the Association for Information Systems

CAIS

ECIS 2010 Panel Report: Humanities-Enriched Information Systems

Jan H. Kroeze
School of Computing, University of South Africa, Pretoria, South Africa
kroezjh@unisa.ac.za

Hugo H. Lotriet
School of Computing, University of South Africa, Pretoria, South Africa

Nehemiah Mavetera
School of Economics and Decision Sciences, North-West University (Mafikeng Campus), South Africa

Mark S. Pfaff
School of Informatics, Indiana University (Indianapolis Campus), Indianapolis, IN, US

Dirk J.v.R. Postma
Department of Education Studies, University of Johannesburg, Johannesburg, South Africa

Kosheek Sewchurran
Department of Information Systems, University of Cape Town, Cape Town, South Africa

Heikki Topi
Department of Computer Information Systems, Bentley University, Waltham, MA, US

Abstract:

This article builds on a panel on Humanities-Enriched Information Systems presented at the 2010 European Conference on Information Systems (ECIS), held in Pretoria, South Africa, June 6–9, 2010. The aim of the panel discussion was to stimulate a meta-theoretical discussion about the relationship between the Humanities and Information Systems in a way opposite to the usual. A lot of research has been conducted on the application of computing in the Humanities, but this panel explored the reverse process of enrichment that takes place. The purpose was to give recognition to work that has already been done in this regard by means of identifying a substantial sub-discipline, but also to inspire more and deliberate research that explores ways to enhance Information Systems by interweaving insights and methods from the Humanities. Such an endeavor may enhance ICT to empower the communities using these technologies.

Keywords: humanities, information systems, social sciences, ontology, philosophy, arts, social psychology, ACM curriculum design

Volume 28, Article 24, pp. 373-392, May 2011

I. INTRODUCTION

Information Systems (IS) is regarded as an interdisciplinary science. Although it mainly focuses on social aspects regarding the development and use of software in organizations, it also deals with programming and algorithms and, therefore, contains elements of mathematical and physical sciences. In addition, insights from the Humanities are as important for this discipline, although this is not always recognized or valued. Many papers, books and articles have been written on Humanities Computing, i.e., the computer-based study of various Humanities disciplines. However, not that much is available on what may be called "Humanities-enriched computing," meaning a Humanities approach toward various aspects of computing and IS.

In this article, Jan Kroeze gives an overview of Humanities-IS synergies. Nehemiah Mavetera introduces "romantic" software development approaches as opposed to mechanistic approaches, referring to ontologies to capture humanistic elements in IS. Dirk Postma, however, cautions against the unreflective proliferation of ontologies based on weak theoretical bases. Heikki Topi echoes this sentiment with lessons learned from a computing-ontology development project. Mark Pfaff changes the focus toward the arts by discussing the implications of some of the more thoughtful approaches to aesthetics in Information Systems. Other panelists look at other human sciences in IS. Kosheek Sewchurran contributes by exploring the current ACM curriculum design using principles from embodied cognition (enaction) and phenomenology. Hugo Lotriet examines the interaction between social psychology and Information Systems in the context of change-focused research.

The aim of the article is to reflect on historical efforts and future opportunities to enrich IS using insights and approaches from the Humanities. By scrutinizing the suggested reciprocally beneficial relationship and synergy between these two groups of disciplines, interdisciplinary work may be stimulated that could grow the IS discipline even further. Embracing conceptualizations borrowed from the human sciences may help the discipline to overcome "the syndrome of refusing to grow" in IS [Monod and Boland, 2007, p. 139]. Therefore, the article is an attempt to accept "our responsibility to look out for the vitality of the field," as suggested by Grover, et al. [2009, p. vii]. The authors trust that the article will provide some inspiration from the arts to help IS scholars come to terms with the pluralism that is inherent in the field [Shoib and Nandhakumar, 2009].

II. HUMANITIES–IS SYNERGIES: FROM COMMENSALISM TO MUTUALISM

Some ICT disciplines, like Information Science, may be regarded as the Humanities branch of ICT, because they developed out of Humanities disciplines like library science. While Humanities computing is the discipline that researches primarily the computer-based study of various Humanities disciplines, Humanities approaches are also present and embedded in other branches of ICT. This sub-field may be called *Humanities-enriched computing*, implying a Humanities approach toward various aspects of computing and IS. Although the symbiotic relationship may currently still be commensalistic (IS is more often used within the Humanities), promising examples show that the relationship could become more mutualistic (IS benefiting as much from the Humanities) [Kroeze, 2010b].

Before venturing into the exploration of Humanities-enriched computing, some clusters of Humanities computing are highlighted in order to clarify the difference. Humanities computing is a wide and well-established field, exploring, i.a., the use of Information Systems to study linguistics and history, aids to edit and enhance electronic art pieces, the investigation of the influence of Information Systems on society and philosophy, and creative research to create e-libraries of theological sources and court cases and interlinear texts.

One outstanding example of Humanities-enriched computing is the current upsurge in the study and use of "ontologies" in Information Systems. Ontology has traditionally been (and still is) a philosophical discipline that has studied the nature of existence. It has fitted into a bigger meta-narrative, such as rationalism. In a certain time and philosophical era, there usually was, therefore, only one correct or current ontology. The plural of this word did not exist, which explains the fact that the plural form, *ontologies,* is not even recognized by the spell checker of a word processor such as MS Word. The shift from ontology to ontologies was probably prompted by the postmodern era in which we live [Kroeze, 2010a]. This article explores the idea of IS ontologies from three different philosophical viewpoints and also looks at other examples of Humanities insights used in IS. Aspects that are not covered, and which should be researched in more detail, are language and IS, and history and IS (compare Kroeze, 2010b).

The next section looks at the mechanistic nature of organizational Information Systems and introduces ontologies to aid in the capturing of humanistic elements in software development.

III. HUMANIZING THE SOFTWARE AND IS DEVELOPMENT FIELD USING ONTOLOGIES

The field of Information Systems Development (ISD) has continuously faced many recurrent problems. In this section, we argue that these problems are reducible to the conceptions that are given to software and Information System Development methodologies. At present, many software products are analyzed and developed using the functionalist paradigm, a notion that has seen a plethora of mechanistic software products being developed and implemented. Strongly attached to this paradigm is the use of the reductionist dogma that also supports systematicity, but becomes skewed when it builds up during system formation. Mechanistic development practices denote the software development field as rational and deterministic. Consequently, when resultant software products are implemented in information systems, these systems exhibit a mechanistic character that eventually limits their usability.

This mechanistic development approach, while addressing the data processing requirements of most organizations, fails to address the dynamic nature of organizational systems. Mechanistic practices overlook the notion that, as representatives of human principals, Information Systems should forge some type of humanistic, non-deterministic behavior. This behavior can only be captured in the software product. Also, since organizations are dynamic, they require dynamic information systems to accomplish tasks and achieve their information-processing goals. Lemmens [2006] noted that dynamic systems are best modeled using predictive models, since they allow for “formal analysis of system behavior.”

Currently, there exists a plethora of software development approaches that are grounded in two development paradigms: the hard systems and soft systems development paradigms. The majority of these approaches follow the structured and object-oriented approaches that are heavily inclined towards the hard systems paradigm. While agile approaches have some resemblance to soft systems approaches especially at analysis stages, they, however, resort to hard systems methodologies at the later stages of software development.

Several attempts have been made to reduce the mechanistic tendencies of software products. Harris et al. [2009] proposed controlled flexible approaches that allow flexibility in the development process but are not too prescriptive. They argued that every software development approach requires some controls. Future research has to look at the controls that should be included in the development methodology and how they should be introduced. Wand and Weber [1990, p. 63] strongly advocated for system development processes that capture both structure (statics) and behavior (dynamics) of the real world. They argued that researchers must use ontologies to provide both “sufficient human-oriented and machine-oriented” descriptions of real world systems [Wand and Weber, 1990, p. 63]. It is against this backdrop that this section proposes an ontology-driven development approach to software and information systems development.

This approach is based on an ontology-driven software development framework that lists a set of ontologies that should be developed and used in each phase of the development lifecycle. We argue that ontologies could facilitate Gregor's [2006, p. 613] dream that Information Systems can be understood only if theory that links the three elements of “the natural world, the social world and the artificial world of human constructions” is found. Currently, many theoretical constructs that are used in the development of IS artifacts do not address this ideal holistically. The hard systems approaches neglect the behavioral aspects of organizations that are usually addressed in the social sciences disciplines. IS researchers are, therefore, encouraged to pull the three disciplines of “natural sciences, social sciences and design sciences” together in order to address the requirements of IS development problems.

Software Development Issues

Several IS researchers and developers, using different theoretical lenses, look at various ways of solving the software development problem. Kawalek and Leonard [1996] discuss the software versus organizational paradox, i.e., the phenomenon that the organization as a holistic real world system is forcibly embedded in a piece of software as a “representation of an organization.” They argue that the piece of software is a partially understood description of the status of an organization at a particular point in time. Unlike software products, organizations are always in a state of flux [Dahlbom and Mathiassen, 1993; Kawalek and Leonard, 1996]. It is therefore important to match the static nature of software products and the dynamic nature of the organization with a dynamic software model. Since organizations are dynamic, software products must be innovative, adaptive, replicable, and evolutionary [Meso and Jain, 2006; Kawalek and Leonard, 1996].

Software evolution is coupled to the fact that change is endemic in organizations. Therefore, “the models, concepts and theories" that were used in software development in the past cannot be used to solve peoples’ development problems in the future [Trim and Lee, 2004, pp. 478–479]. Kawalek and Leonard [1996, p. 189] lamented the IS fraternity’s failure to develop methods and practices that produce “instantly adaptable software that is able to support radically changing demands on a series of fast developing platforms and integrating with a series of end user developments." In addition, they argued that software development requires approaches that recognize the duality of organizational context and that of developing software products. Lemmens [2006] further motivated for a paradigm shift in the way information systems are developed. He argued that current systems are too data-centric and that developers should move to functional centric methods of developing systems. At a functional level, processes capture the behavioral aspects of systems, thereby neglecting the use of instances of data as system behavioral representations. In a bid to change the thinking and focus for IS developers, Hohmann [2007, p. 19] advocated for the development of intuitive systems that are easily understood by humans and at the same time increasing the productivity gains from their use.

Transition from Mechanistic to Romantic Information Systems

Therefore, ISD must focus on the need to improve the usability and adaptability of current Information Systems. Mechanistic systems allow efficient and effective computation and structuring of data but still remain very syntactic. Some users of such systems need to engage in another cycle of interpretation, after the computation cycle, in order to make sense of the results. Systems of this nature do not provide meaning and context of what has been processed to the user. In Tarnas's [1991, p. 266] words, the material particles captured and represented in these systems possess "neither purpose nor intelligence." These mechanistic systems have been blamed for their failure to capture and maintain organizational culture, provide meaning to concepts in use (semantics), pragmatics, and social context. It is important, therefore, to propose a shift to the development of "romantic" information systems.

In the context of this discussion, romantic information systems are not only limited to the dictates of syntactic machine representations. These systems are based on the romantic worldview that considers the "world as a unitary organism" as sharply contrasted to the rational atomistic view of the mechanistic world [Tarnas, 1991, pp. 366–367]. The idea for romanticism in Information Systems is supported by Hohmann [2007, p. 18] who calls for a "pluralisation of our culture and the humanization of technology." In his vision, he sees a future that demands technologies that stimulate creativity and inspire thoughts, thereby reconciling the "contradictions between technology and art" that characterize the modern era.

Toward an Ontology-Driven IS Development Approach

Industrial and academic researchers currently focus their attention on the semantic properties of ontologies to improve the development of Information Systems. This endeavor could again prove to be another futile exercise, if they are not considering how ontology properties could be introduced methodologically in the development of software products. Isabella [1990] notes that many research studies focus on the design and development of concrete and observable aspects of ontologies in Information Systems, but very few pay attention to the identification and understanding, the interpretations and cognitions associated with the use of ontologies in Information Systems. Also, despite all the efforts to use ontologies in Information Systems, there is still no clearly defined purpose of IS ontologies. This deficiency is blamed on the lack of a clear distinction between philosophical ontologies and IS ontologies. Zúñiga [2001, p. 188] believes the IS discipline is "either not equipped to advance general ontologies or not employing the right methodology or theoretical approach" to the use of ontologies in this field.

What is Ontology?

Ontology has attracted many definitions when used in IS. The term was coined in the early 1990s by Tom Gruber and his research team [Neches et al*.*, 1991; Gruber, 1993] at Stanford University. The definition that seems to be widely adopted portrays ontology as a formal specification of a shared conceptualization [Guarino, 1998; Gruber, 1993; Studer et al., 1998]. Guarino et al. [1994] also regard it as "knowledge about a priori structure of reality," implying that, when one chooses a particular intended model for a logical theory, one has to make "implicit assumptions about other models that are compatible with the chosen one." In Information Systems, ontology can be regarded as a "formal language designed to represent a particular domain of knowledge" [Zúñiga, 2001, p. 187]. In this field, it plays a functional role since it is almost designed for specific purposes.

Its use in Information Systems (IS) has been increasing gradually. It is used in the fields of semantic Web and database systems [Gruber, 2008], library sciences [Ding and Foo, 2002], routing systems [Winter and Tomko, 2006], Information Systems and software development [Mavetera, 2007; Aβmann et al*.*, 2006; Dristas et al*.*, 2005; Corcho et al*.*, 2006], to mention but a few. Ontology has also been used in accessing legacy resources [Simonov et al., 2004] through an ontology-driven natural language Web-based access system. More interestingly, Soffer et al. [2001] used ontologies to bridge the gap between business requirements and off-the-shelf Information Systems software capabilities in order to adapt the business to software capabilities. As is evident from these few examples, many fields are already using ontologies and software development is just one specific application area. However, it must be noted that ontology's structural components, semiotic requirements and its representation in Information Systems continuously constitute a subject of debate.

Of the two parallel streams, that is, semiotics and formal logic, that can be used to describe ontology, our discussion uses the semiotic stance [Sowa, 2000; Stamper, 1992] to build an argument that posits ontology as an artifact that can solve, among others, the current problems of linguistic communication during software and systems development and the capturing and representation of human and softer characteristics of organizations. Since IS ontologies are confined to that which can be represented, it must be noted that subjective things such as feelings, even though they exist, cannot as yet be represented especially in automated machines. This is not a limitation to the development of romantic software products but a shortfall attributed to the technology used for representation.

Why Use Ontologies in IS?

There have been several calls for new methods of developing software products, with Cretu [2010] proposing a business process oriented software (BPOS) methodology and Ben Sta [2010] advocating for a software development method that merges ontologies with currently "existing methods, techniques and tools" that are used during the analysis phase. Ben Sta [2010] argued that the merger can improve significantly the process of software development at all stages of analysis through to maintenance, by facilitating the faithful translation of user requirements into object models that are used to develop the system specification. Mavetera [2007] proposed the Ontosoft framework that also, like Ben Sta's [2010] framework, positions ontologies at the center of an automated software development case tool. Hofferer [2007] discussed a semantic interoperability approach that combines meta-models and ontologies. In this approach, ontologies complement meta-models by adding semantic expressiveness to business models. This built-in meaning is important when integrating business models for different organizations. More to it, ontologies have long been accepted as artifacts that can improve the software process [Falbo et al., 2002], while Wand and Weber [1990, p. 69] argue that an "ontological approach to understanding and formalizing information systems concepts provides"…"the rudiments of a theory of the deep structure of an IS." The lack of such a theory deeply undermines research and development of Information Systems products that have human characteristics.

How Should Ontologies Be Used in IS?

We are going to provide a solution set to current problems in software development. In this solution set, which is a framework of ontology components (Table 1 below), the ontology is positioned as an artifact that can be used during the development of Information Systems and at run-time stage to decrease the semantic gap currently existing in Information Systems. The framework depicts components of the ontology that can be looked for by any would-be system developer in a bid to make the resultant system romantic. This analysis is a quest for finding and defining an "effective problem representation" [Hevner et al., 2004, p. 83] and as a construct is crucial for software developers to effectively design software solutions. This framework was originally developed in Mavetera and Kroeze [2010].

Discussion

The purpose of the analysis model is to capture the triplet, domain model, business model, and the requirements model. The business model is tasked with capturing the organization's rules of business and, lastly, the requirements model, which models the system specification, is tasked with capturing the functional and non-functional system requirements [Aβmann et al., 2006, p. 254]. The software and Information Systems development processes require an ontology driven analysis model that is made up of domain, process or method ontologies to capture the domain, business and specification models of the system to be represented as a software model.

On the other hand, the design model is the architectural model of the system and, at this stage, it should capture the system from the designers' viewpoint but should still be platform-independent. Lastly, the implementation model is gradually populated with platform specific details as discussed by Aβmann et al. [2006]. There is always a gap between the "kinds and forms of the domain knowledge in the domain model and the content and form of software assets" that are constructed. To reduce this, Falbo et al. [2002] proposed an infrastructure specification that, with its semantics as captured by the domain model, could be used as input to the implementation phase of the software development process. Wand and Weber [1990, p. 63] advocated for a complete translation of the analytic model attributes through to the implementation model using ontologies.

Table 1, column 2 depicts the IS development problems that need to be addressed, and column 3 the ontologies that can be used in order to address the IS problem. It must be noted that, for each requirement, developers are urged to develop ontologies that satisfy the said requirement and store them in a software development environment. These ontologies can then be used as the basis upon which the human aspects of organizational information systems can be captured, stored, and reused during software development. For example, in order to capture and maintain the business and domain model characteristics of organizations from analysis to implementation, domain ontologies can be developed and used during software development.

Table 1: Humanizing IS Products Using Ontologies (Adapted from Mavetera and Kroeze, 2010)		
Issues	**Software development problems**	**ISD ontologies required**
1	Capture and maintain business and domain model characteristics from analysis to implementation.	Use domain ontology as software model, capturing the knowledge valid for a particular type of domain.
2	Capture possible life states of a system.	Ontology is used as an intensional model of the system.
3	Capture system requirements—requirements specify the processes that run in organizational systems, that is, capturing the specification model.	Use process, method, task, or activity ontologies. These ontologies capture the knowledge and reasoning needed in performing a task, i.e., the "what" and "how" of doing a particular task, for example, requirements gathering task, or software design task. They also capture the knowledge and reasoning needed in performing a task. Concepts used in task/method/process ontologies should be derived from the domain field so as to reduce the risk of losing the descriptive nature of the method ontology in the domain.
4	Capture the descriptive analysis model—the analysis model is a descriptive type of a model that conforms to the open world assumption.	Domain, process, or method ontologies.
5	Capture behavioral attributes of systems, i.e., static and dynamic attributes. Software development must allow for modeling of both the static and dynamic states of a system.	Status ontologies can capture the static (change in form of existence) and dynamic (time dependent) aspects of organization. These ontologies represent the status of an artifact. They argue for a world with artifacts that exist and do not change their form of existence (static) and also another class of things that change with time (dynamic). As such, dynamic ontologies are used to abstract the behavioral characteristics of a system.
6	Capture system behavioral aspects.	Intentional ontologies—these ontologies are intended to model the softer aspects of living things, which can have beliefs, desires, and intentions (BDI). In this category, the human aspects of living things are modeled and examples of such ontologies are aspect, object, agent, and support as stated in Ruiz and Hilera [2006]. These types of ontologies are also meant to model ascriptions of intensions to actors in a system.
7	Capture organizational culture and context.	Social ontologies describe the organizational structure and the interdependencies that exist among the social actors in these organizations. At a high abstract level, they include concepts such as actor role and responsibility to mention but a few.

There are, however, several practical applications where ontologies can be used during software development. This framework, therefore, is not exhaustive. As Sarantakos [1997] noted, a framework like this is supposed to emerge from experience and must be continuously revised and corrected through several research studies. Most importantly, it must not act as a "blinder or straitjacket." It must be directed and fine-tuned to serve the needs of software developers. The framework as presented here lacks some application context in terms of a formalized way of using it. Therefore, it requires an ontology driven software development approach coupled with a methodology for it to be used in every day software development process. The use of this framework may start reintroducing the ever elusive human elements in information systems.

The next section warns against the tendency to draw on complex theories in an eclectic way during the design of IS ontologies and pleads for a more responsible approach with regard to the unreflective proliferation of ontologies and the attempts to create a meta-ontology.

IV. A CRITICAL, PHILOSOPHICAL REFLECTION ON INFORMATION SYSTEMS ONTOLOGIES

The design of Information System ontologies (ISOs) is mainly motivated by a pragmatic concern to develop a meta-language through which the proliferation of Information Systems could be mediated. Smith [2003] indicates that such an approach cannot distinguish well between good and bad conceptualizations. He claims that ISOs have to learn from the long tradition of philosophical ontology in order to relate these conceptualizations to real features of the world. Drawing on Quine, Smith states that such conceptualizations are judged to be good or bad particularly in relation to a scientific conception of the world which is the most reliable appeal to an independent reality. This section develops further the value that philosophy may have for IS through an alternative investigation of philosophical ontology. The pragmatism of ISOs and the realism of Smith are contrasted with a performative ontology which makes it possible to open up the contestable nature of technological design.

A conception of ontology within actor-network theory (ANT) is used to show that ISOs do not simply promote translatability or represent reality but that they contribute in important ways to the performance of realities. This section cautions against unreflective ways in which ontologies are proliferated and meta-ontologies are developed. Designers of ontologies should be critically aware of the politics of ontology and of the real effects of their work which may not benefit everyone equally.

The development of ISOs aims to address the bewildering plurality of information systems through the creation of a meta-language and set of categories which could act as a reference point for the translation of regional languages. Through conceptual modeling and the utilization of appropriate categories, this meta-language promotes translatability and contributes toward harmonizable Information Systems. This concern is mainly pragmatic [Smith, 2003] since it aims at a workable solution out of the looming crisis of incompatibility and incongruence. This mission is a very important project because it attempts to build bridges between different Information Systems.

According to its pragmatism ISOs are not concerned about the relation between the concepts and an underlying reality, but with the “workability” of things within practices. The question is not whether a meta-language describes a deeper layer of reality, but whether it succeeds in mediating the different languages and whether translatability could be achieved. Insofar as the conceptualizations relate to domain-specific languages, a meta-language aims to capture adequately the meanings users attach to concepts.

The realist ontology Smith [2003] proposes aims to represent what is out there and to remain true to things as they are. In this sense it claims to be a neutral register since it only captures what already exists. The basic assumption is that reality consists of discrete entities that could be captured accurately through concepts and categories. While different information systems use different languages, they all represent the same underlying reality. It is, therefore, possible for a meta-language to appeal to this underlying reality and to achieve translatability. Translation is then a process where both languages are related to each other on the basis of their translatability into the meta-language. Different languages represent different perspectives on the same reality which could be mediated if one could step outside these relative views into a more objective view of the real.

This contrast between pragmatism and realism hides the fact that both perform reality. The performance of reality is underlied by ontological commitments [Quine, 1953] which entail normative notions of reality. ISOs can never be seen as a mere pragmatic or as a neutral description of reality: they both generate an idealized reality. Whether ISOs attempt to represent reality or to find a pragmatic solution, reality is being performed. The realist ontology does not simply describe and represent reality, but also creates categories, identities, and relations. This constitutive process is also the case with an ISO based on pragmatic assumptions. Whether the categories and concepts are seen in a more instrumental sense as contributing to some notion of success, they have implications for the way in which reality is generated. The very domain-specific concepts on which an ISO is based are not merely tools in the hands of their users, but they perform the reality of the domain. In a health information system the realities of “psychiatrist,” “patient,” and “diagnostic” are being performed [Bloomfield and McLean, 1996], as are the realities of “researcher” and “university” in an Information System in a tertiary institution [Scott and Wagner, 2003].

These examples illustrate how reality is being performed through our practices, networks, or assemblages. These assemblages are constituted by multiple entities (humans and nonhumans) that relate to each other in many ways. In this way the reality of a hospital, a laboratory, a state, or nature is being performed. Any of these realities cease to exist if they are not continually performed by various agents. The realities are performed through the agential role of multiple and heterogeneous entities such as instruments, technologies, material objects, intentions, plans, concepts, categories, procedures, policies, and desires.

Therefore, ISOs have important implications for reality and could never be seen as a neutral description of the world or as a merely efficient system. Designers and researchers should go beyond the realist ontology of representation and the pragmatist ontology of workability to recognize the political way in which realities are being performed. This politics refers to the potentially problematical ways in which the categories [Bowker and Star, 2000], agencies, identities, and resources are identified and distributed.

What makes political awareness more pertinent is the powerful way in which technologies contribute to the irreversible [Callon and Latour, 1981, p. 301] and immobile [Law, 1986, p. 241] of the categories and identities. The embeddedness of particular ontological assumptions in an ISO meta-language has far-reaching and potentially problematical implications for the way reality is being performed as testified to in the following statement:

> The information sciences have this century grappled with new ways of configuring, storing and retrieving information, as fundamentally novel as was the printing press in its day…. This new infrastructure has powerful ramifications, comparable to the railroads …. or electricity … infrastructures that respectively accompanied the first industrial revolution and drove the second. Because new information infrastructures fundamentally change both work practice and knowledge, they also inscribe a moral order. They do so by allocating resources …. structuring markets … and affecting the rhythm of daily life [Bowker et al., 1996, pp. 345, 346].

This argument implies that ISOs do not perform a single reality, but that many realities are being produced. Medical research illustrates this phenomenon where Mol [2002] explains how different diagnostic and treatment practices produce different realities of lower-limb arthrosclerosis. It is an error to assume that, since one concept (lower-limb arthrosclerosis) is being used, there must be a single underlying reality. She shows how one reality of lower-limb arthrosclerosis is performed in the consulting room, another in the pathology laboratory, another in the radiology department, another through an angiograph, and still another reality in the operating theatre. It does not imply a relativistic fragmentation of realities, since comparisons, translations, and associations are always possible. Overlaps exist, but at other times contradictions between these different versions of reality come up which prevent them from being added up to a single reality.

The notion of a universe (single reality) is now replaced with that of a pluriverse, which is different from the notion of a multiverse which postulates the separate existence of different realities. There is not one reality as seen in the tradition of philosophical ontology, which we could adequately capture through our linguistic categories. We also should not see reality as fragmented. In order to refer to such a position, Law [1999, p. 12] uses the notion of *fractal* to refer to any network that is "more than one and less than many."

If we accept this ontology, then we should be able to trace how ontics (realities) are being generated and sustained through ISOs. We should also be able to establish what the effects of one reality are on the other.

This section briefly indicated how philosophical reflection could contribute toward the development of Information Systems ontologies. This idea demands that designers become aware of their political work, since they shape reality in ways that could have liberating or oppressive consequences for humans, things, and nature. It demands that they treat the ontologies they design not as "matters of fact" to be accomplished, but as "matters of concern" [Latour, 2004, pp. 231, 232] which require insight into the ways they are generated and their implications for the entities that are enrolled. The next section may be seen as one attempt to use a proper conceptual foundation related to ontologies in computing in general and Information Systems in specific. It aims to improve our understanding of the field of computing and the relationships between subdisciplines of computing. It highlights the importance of building any scholarly and pedagogical work related to concept structures on a strong foundation that relies on work in philosophy, cognitive science, artificial intelligence, and computer science. It also suggests a connection between Humanities and the key areas in which Information Systems provides unique value within computing.

V. ONTOLOGY, ONTOLOGIES, COMPUTING ONTOLOGY, AND CONCEPTUAL MODELING

The challenges of classification and categorization have for a long time been recognized to be in the center of any process that focuses on understanding a specific domain or an area of the world. In the work of information systems and information technology practitioners, these questions are most often addressed in various types of data and information modeling processes and other approaches that represent a specific domain of discourse using a structured, well-defined grammar [see Weber, 1997, p. 75].

Weber and Wand, together with many of their students, have for a long time been developing a body of research that demonstrates the contributions of ontological theories (and specifically the work on this topic by Bunge [1977 and 1979]) to the understanding and development of information systems as representations of "other real-world systems" [Wand and Weber, 1995; Wand and Weber, 2002]. Largely based on Wand's and Weber's work, Bunge's thinking had a strong impact on research on Information Systems also more broadly, as demonstrated by about 150 articles and conference papers in Information Systems referencing Bunge [1977] and Bunge [1979].

These papers deal with topics as widespread as end-user query development [Bowen et al., 2009], accounting principles [Chou and Chi, 2010], and business processes [Ghattas and Soffer, 2009; Recker et al., 2009], in addition to the fundamental work on conceptual modeling [e.g., Parsons and Wand, 2008; Burton-Jones and Meso, 2008; Gemino and Wand, 2005; Shanks et al., 2003]. Despite justified observations regarding the shortcomings of the Bunge-Wand-Weber ontology [Rosemann and Wyssuek, 2005; Allen and March, 2006], even a cursory evaluation of this work demonstrates that the application of Bunge's work on ontology to the field of Information Systems provided the field with a significantly stronger theoretical foundation. It is likely that a future evaluation of our discipline will conclude that Bunge's work as a philosopher contributed in a fundamental way to the maturation of Information Systems as a field of study—definitely a substantial example of foundational work in Humanities affecting Information Systems.

The practice of creating ontologies as formal representations of concepts (and thus, knowledge) within a specific domain has become commonplace in a number of information fields, such as artificial intelligence, Web science, and semantic Web, library science, and information and enterprise architecture. Gruber [1995, p. 907] calls these "ontologies used for knowledge sharing" and defines *ontologies* as explicit specifications of a conceptualization. Gruber explicitly acknowledges the origins of this meaning of the word *ontology* in philosophy, and it is clear that the structural characteristics of ontologies for knowledge sharing are (or at least should be) based on the deep understanding that work in the subfield of ontology in philosophy produced over the past hundreds and thousands of years. The quality of concept structure ontologies is not only dependent on the domain expertise embedded in them, but also on the quality of their metastructure which, in turn, is strongly affected by how well the philosophical principles are applied in the development of the underlying structure.

Information Systems as a field has the potential to benefit significantly from one specific process to develop a domain ontology for computing, first to support the needs of computing education with the later potential to expand the uses of the ontology to other uses (research classification, for example). The Computing Ontology Project [Cassel et al., 2007a, 2007b) is a joint project by the ACM Education Board and the IEEE-CS Education Activities Board, and it intends to "enumerate all of the topics related to any aspect of the computing disciplines" [Cassel et al., 2007a, p. 519]. The primary motivation for the project came from the need to support the efforts of maintaining the bodies of knowledge for various computing disciplines in the joint ACM/IEEE-CS and ACM/AIS curriculum recommendation projects. Of particular importance in this context is the ability to highlight the similarities and differences between various sub-disciplines of computing in terms of their use of the same concepts. In addition, the project recognizes the importance of the maintainability of the concept structure, particularly in a field that is changing as rapidly as computing is.

It soon became clear, however, that if the project is successful, the continually evolving concept structure could be very helpful for other types of computing research and education classification needs. At the same time, it became equally evident that the scope and the complexity of the project are enormous and that for the project to be successful, it needs to be based on a strong understanding of the theory of understanding and representing the relationships between the concepts included in the structure. Here, the contribution of Humanities becomes highly valuable, particularly the work related to ontology in philosophy that focuses on the basic categories of being and the specification of distinctions between categories.

The computing ontology project is still searching for the best structure for representing the concept hierarchy in computing, but the project made good progress in identifying and tentatively organizing the core concepts using a diverse set of sources, including the ACM/AIS/IEEE-CS curriculum reports, several national model curricula, and the ACM Computing Classification System, among other sources. The current results of the project are available at http://what.csc.villanova.edu/twiki/bin/view/Main/OntologyProject. The search for the representation structure for this particular computing ontology will bring us back to the work that has been done in Information Systems related to representing a domain: Wand's and Weber's introduction of Bunge's thinking regarding ontology as an important theoretical foundation for building representations (such as ontologies as concept structures) was instrumental in moving research within this area of study forward in Information Systems, to the extent that IS research on representations is currently theoretically sound and practically applicable. When the computing ontology project is seeking for a solid foundation for its representation structure, it is very likely to be informed by Information Systems research in this area. This research, in turn, has been strongly influenced by work in Humanities.

The next session also looks at the ACM curriculum, but from a social sciences perspective. It explores the ACM curriculum design using the work of existentialist phenomenologists Martin Heidegger, Merlou-Ponty, and Maturana and Varela. It attempts to do a principled assessment using principles from embodied cognition (enaction) and phenomenology.

VI. THE LATEST ACM2010 CURRICULUM: WHAT DOES A LACK OF SOFTWARE DEVELOPMENT PORTEND FOR THE IS GRADUATE?

To discuss Humanities enrichment of IS, this part of the panel discussion first looks at the possibilities for getting theories to embrace the as-lived human condition more correlatively. The discussion starts by describing the ideals of embodied cognition and contrasts them to the ideals of rationalism that emerged from the natural sciences through the accumulated tradition of positivist research practice. Second, the discussion connects the potential pursuit of this ideology to the kinds of benefits that are possible by reviewing the 2010 ACM curriculum. Before proceeding, it is worth stating that research efforts in the natural sciences, human sciences, and sciences of the artificial are all attempts to explain, predict, infer, and/or describe some phenomena to bring forth new understanding.

The 2010 release of the ACM curriculum embraces a new format to guide undergraduate curriculum designs. The curriculum now has core knowledge areas and specialized knowledge areas. The designers say that they settled on this structure to encourage further technical development of the specialized areas. The core knowledge areas imply that they are central to the development of the specialized areas. It is very likely that the panel had to balance divergent interests of employer needs, pedagogical principles, the maturity of current knowledge areas, and the key graduate abilities that guided the curriculum design process. Throughout this process the panel had to have been informed by some idea of how the graduates would execute their daily duties. Was this a Humanities-enriched perspective? We argue that their choice to make development non-core is an example that illustrates that a Humanities-enriched perspective seems to be lacking. Had they had a Humanities-enriched perspective, they would have been more aware of how their subsequent decision to make development non-core would affect graduates.

It is argued that Information Systems is a science of the artificial which is concerned with the study of the artifacts that arise from human enterprise. The term *science of the artificial* comes from Simon's terminology [Simon, 1996]. Generally the sciences can be placed in three interconnected categories [Strasser, 1985]. One category is for the natural sciences, which study naturally occurring phenomena. Another category is for the human sciences, which focus on human beings as the object of study and include disciplines such as psychology and sociology. Students thus study more than just the technological system, or just the social system, or even the two side by side; in addition, they study the phenomena that emerge when the two interact. IS is thus a discipline that is concerned with artifacts (manufactured objects—plans, predictions, prescription, explanations of usefulness, concepts, etc.) that are brought about or "caused" by human agency. Consequently, graduate students have to have a way of coping with and understanding the complex world of lived experience from the point of view of those who live it.

The ACM 2010 joint curriculum task force proposes that the development stream is not a core knowledge area that needs to be taught in an undergraduate degree. In the argument against this decision, reasons will be presented that tend toward the problems which will arise when graduates are not able to understand the complex world of lived experience from the point of view of those who live it. Generally IS professionals and academics realize the majority of students emerging as IS graduates are not nurtured to pursue careers as developers; instead they are more likely to focus on the analyst, modeling, and managerial career outcomes. Doing away with development altogether still concerns some of us tremendously because we realize that development knowledge is crucial for understanding and appreciating analyst, modeling, and managerial outcomes. The argument is primarily based on how integral some sort of development or configuration project is to achieving the main outcomes that IS programs try to develop in IS graduates because all new knowledge will emerge from the paradigm of understanding with which a graduate leaves the program. The concern is that by defining development as non-core there will be no obligation to engage in the development of reflexive knowledge. It thus seems likely that there will be an overemphasis on instrumental knowledge.

It may be that the best way to orient this discussion is to ask: How does the knowledge we equip our students with enable them to respond correlatively to various challenges in their professional lives? Alternatively, we could ask: How do graduates respond appropriately to the messy ill-defined contexts in which they will be plying their IS competence? Will they seek out characteristics of the context and then select the appropriate practices from the knowledge areas? Probably not!—A Humanities-enriched perspective will argue [Merleau-Ponty, 1945; Maturana and Poerksen, 2004]. It is more likely that an image of the context will construe what is noticed. Furthermore, it is widely known that learning occurs through experience; learners first need to undergo a particular experience and then, upon reflecting upon that experience, extrapolate learning from it. Learning of this nature is important to new practitioners, for once they enter the world of practice, no matter how hard they try to apply theoretical criteria or use advanced analytic techniques, they confront technical, cultural, moral, and personal idiosyncrasies which defy categorization. Often such experiences make graduates feel under-prepared and alienated by the profession they expect to typify.

Is it not prudent then to intervene during these fragile stages of competence development? Over the years it seems many academics have grown to realize that students need the opportunity to try out their conceptual knowledge so that it becomes contextual or grounded, in a world, that it becomes "do-able" [Scott and Sewchurran, 2008; Sewchurran, 2008; Sewchurran and Scott, 2009]. It may be that the concept of capstone courses emerged for this very reason.

In fact, reliance on conceptualization alone may just create instrumental technicians who think that through a rigid adherence and reliance on methods, specific processes and/or artifact production alone they will be able to successfully:

a) Improve organizational processes
b) Exploit opportunities created by technology innovations
c) Understand and address information requirements
d) Design and manage enterprise architecture
e) Identify and evaluate solution and sourcing alternatives

A Humanities-enriched curriculum would argue that such an ideology is retarding, and it may be the case that this ideology exists already and is currently implicated in many of the wider problems reported in the literature. The agility debate, for example, is a prominent one. Perhaps we should ask: Is agility a practice for novices or is it a maturation point once practices become embodied? Is the business and IS alignment discussion a result of over-emphasis on artifact production instead of business outcomes? Without the means to do a project which necessitates some development, IS programs will probably produce graduates that perpetuate the problems that arise through instrumental ideologies.

The splitting up of the knowledge areas of project management, analysis and design, development, etc. is for pragmatic reasons only. Without the basis (ability to develop or configure software) to combine them in a capstone process, it is superficial to think any competence is being developed. It is more likely that we will develop graduates who are retarded by the paradigms they become socialized into. In summary, without any development or configuration project, it can be inferred that the following view of professional work is promoted [Donnelly, 1999]:

a) Practices are founded on propositional / representational knowledge.
b) Practices can be exhaustively analyzed, meticulously planned and executed without the need to consider contextual circumstances and adjust behavior and intervention accordingly.

A lack of development or configuration project experience will make it almost infeasible to encourage learning, reflective practice, and sense making [Scott and Sewchurran, 2008; Sewchurran, 2008; Sewchurran and Scott, 2009]. With development defined as non-core, many undergraduate programs will simply ignore any form of development because it is difficult to teach. Furthermore, some studies indicate students do not enjoy programming. The extent to which programming leads to deeper learning is not appreciated in the ACM curriculum. The ACM task team's decision may affect the quality of graduate that emerges from programs as academics try to adhere to the new ACM guidelines. We would specifically find it difficult to get students to embody the following characteristics:

a) Realize that reality for each individual is a unique experience and effort is needed to re-orient or understand orientation of individuals
b) Be capable of multiplicity in thinking (accepting multiple versions of purposeful effort) instead of binary thinking (right or wrong)
c) Be aware that perfect communication is impossible and that communication is transformative rather than the passive transfer of information
d) Be suspicious of method adherence (method-ism or instrumentality) because it is not a sustainable effective strategy
e) Be aware that knowledge necessary to perform useful work cannot be and is not a body of information to be learned and learned once only

In summary, this panel argued that a key factor contributing to what amounts to a lack of insight into the effects of the new curriculum is not enough Humanities enrichment. Education could learn from the arts where the digital divide is being overcome by a convergence of IS and aesthetics, as discussed in the next section.

VII. AESTHETICS AND INFORMATION SYSTEMS

C.P. Snow's *The Two Cultures* lecture published in 1959 argued that there was a "gulf of mutual incomprehension" [Snow, 1959, p. 4] between the culture of arts and Humanities and the culture of science. Snow's complaint was that artists had no understanding of science, nor did scientists understand anything about the arts. However, a convergence is suggested by the fairly recent appearance of the concept of aesthetics in the Information Systems (IS) literature. It is often applied in unskillful ways, ranging from a catchall term for the sensory aspects of user experience, to simply a lofty synonym for "pretty." Fortunately, some researchers have done ample justice to the rich philosophical contribution of aesthetics in Information Systems. A review of some of the more thoughtful approaches to aesthetics in IS may shed some light on whether we are seeing a movement toward unification, or only an asymptotic approach in which the two sides never quite meet.

Snow's argument was controversial (and is still discussed today), though the notion of some kind of incompatibility between the Humanities and sciences was not new. In fact, almost eighty years earlier, in the same Rede Lecture series at Cambridge University, Matthew Arnold's lecture "Literature and Science" questioned the relevance of a classical education in an age of tremendous scientific advances [Arnold, 1960]. At the heart of this argument is the question of whether or not there are fundamental differences in the ways we think about the sciences and the Humanities. Maurice Merleau-Ponty [1964, p. 3] called for the formation of "a new idea of reason," suggesting that reason based on logical and mathematical rationality may be ideal for describing the atom or predict planetary orbits, but it is insufficient to understand the expressive meaning of a painting or a poem. He continued: "Expression is like a step taken in the fog—no one can say where, if anywhere, it will lead." It is indeed quite unlike the development of algorithms, which are designed to lead us reliably to the same place, every time.

But how logical and deterministic is an information system? IS distinguishes itself from computer science by keeping people and organizations in the picture at all times, making it truly a social science. Beyond the software, IS is very interested in what people *do* with the software, and with each other. The field of IS, therefore, is holistically-minded. IS research employs mixed methodologies to consider technology, information, people, communities, organizations, environments, history, and more. These are all lenses through which people *experience* Information Systems, and it is this notion of experience that brings us to aesthetics.

Aesthetics is a branch of philosophy concerned with the values and judgments assigned to experiences, based on the human faculties for sensation and perception. These judgments are generally focused on identifying things that are pleasing, especially *beauty*. Aesthetics studies the many different ways in which people perceive things as beautiful, and what meaning such beauty has within and between people. These judgments occur at the visceral, behavioral, and reflective levels, coincidentally aligning with Donald Norman's three levels of the cognitive/emotional system [Norman, 2004] which he uses to describe how people interact with everyday tools and products. Therefore, anywhere that sensations or perceptions of information systems are at stake, aesthetics is highly relevant.

The meaning of beauty within an individual or a community is essentially a discussion of *taste*. Taste is a socially constructed manifestation of shared aesthetic values. From the perspective of the social construction of technology [Bijker et al., 1987], the notion of taste is very relevant to the social context in which technology develops. A community's taste for IS shifts and cycles like tastes in fashion, music, or food. Therefore, an IS student aiming to become an IS practitioner should have a solid understanding of how people determine what they find pleasing and good (and not strictly in a hedonistic or pleasure-seeking sense, but how we can expect people to be oriented toward certain features and averse to others).

Where could aesthetics enhance information systems? For the end-user, the pleasure, usability, and effectiveness of a system all have aesthetic considerations. For developers, aesthetics can impact the methodology of designing and implementing IS, including the pedagogy and training of future IS practitioners. Fully exploring all of these areas is beyond the scope of this article, but several issues can be identified that will hopefully spur continued research and discussion.

Beginning with pleasure, we may ask: If we are stuck using Information Systems, is there a mandate to make them fun to some degree? Is *fun* antithetical to *work*? Tiger [2000] describes four distinct types of pleasure, including physio-pleasure (physiological, from the senses), socio-pleasure (sociological, from interacting with others), psycho-pleasure (psychological, the cognitive and emotional experience of pleasure), and ideo-pleasure (ideological, from valuing and judging experiences). All of these are viable interdisciplinary opportunities for exploring and enhancing pleasure in IS.

Turning to usability, a critical principle in IS, much research indicates a significant influence of aesthetics. Noam Tractinsky [1997] found compelling connections between aesthetics and the usability of automated teller machine

(ATM) interfaces. Bolter and Gromala [2006] describe the twin aesthetics of transparency and reflectivity in computer systems. *Transparency* is when the computer system seems to disappear as the user works. Examples are seen in discussions of flow theory [Csikszentmihalyi and Csikszentmihalyi, 1988] and presence [Sheridan, 1992]. *Reflectivity* refers to whether the user is critically aware of the computer system. It involves how users notice features and reflect upon their relationship with the system. However, despite these empirical findings about the relationship between aesthetics and usability in a variety of contexts, it is still an open question whether the aesthetics that are good for usability are consistent across cultures. Tractinsky [1997] found surprising similarities in the impact of aesthetics on usability in studies in Japan and Israel, but any determination of aesthetic universals in IS is still a long way off.

With regard to effectiveness of Information Systems, there is a long-standing debate on whether the look and feel of a system is an enhancement or a distraction. An example may be found in the field of information visualization, which is an interesting intersection of art and engineering. Edward Tufte [1997] provides guidance on the balance between form and function, asserting that a visualization should be *necessary* first, and *pleasing* second. Norman [2004, p. 17] asserts "[a]ttractive things work better," and suggests emotion as the link between aesthetics and effectiveness of products. However, as the focus shifts toward emotions, the discussion moves over to that intersection of computer science and psychology known as *affective computing* [Picard, 1997]. But is there such a thing as *aesthetic computing*? The answer is yes.

The field of aesthetic computing [Fishwick, 2006] is exploring the channels by which aesthetics could migrate into IS practice. Roger Malina [2006] describes several opportunities identified in the literature, ranging from metaphorical appropriation of artistic concepts by scientists, to collaboration between artists and scientists as peers, to artists trained in science sufficiently to develop their own new information systems. If an IS design guideline is derived from some artistic tradition, does that make it an aesthetic concern? Gestalt psychology shows us how the human mind likes to organize what it sees (alignment, similarity, and so on), whether in a painting or on a computer screen, so surely the basic visual guidelines taught to budding interface developers are quite similar to the principles taught to art students, though by substantially different means.

This discussion lastly deals with the role of aesthetics in pedagogy and training. Returning to the Two Cultures debate, each side scoffs at the other for their ignorance of what they believe to be fundamental knowledge: physicists are ridiculed for their ignorance of Shakespeare, while historians are mocked for their ignorance of thermodynamics. To what extent should aesthetics fit into an Information Systems curriculum? Should IS students be expected to take a philosophical seminar in aesthetics? Probably not (though some do). Would it be good for aesthetics to be properly understood in IS, or at least not used pejoratively? Absolutely. Progress is being made in this direction, as more IS schools offer both bachelor of science and bachelor of arts degrees, and practitioners with master of fine arts (MFA) degrees in art and literature appear among their faculty and administration.

Art and information systems have already met. They may not understand each other, but they are working together. Practitioners apply their aesthetics to Information Systems whether they know it or not. The holistic viewpoint of IS encourages us to reach beyond the technologically deterministic aspects and embrace the social and psychological factors of IS, no matter how unusual or unpredictable they may be. Aesthetics helps us appreciate the richness, variation, and discontinuities one finds when examining any population doing anything of interest.

Some of the reflections above have already touched on aspects in the social sciences too. This idea is now taken further by examining the interaction between social psychology and Information Systems in the context of change-focused research. The validity of the differentiation between the Humanities and the social sciences is also questioned.

VIII. HUMANITIES-ENRICHED IS RESEARCH: A COMMENT ON BOUNDARY ISSUES IN IS RESEARCH

This contribution focuses on various boundary issues that arise when the topic of "Humanities-enriched IS research" is approached from a social psychological perspective. The contribution is structured in the following manner: (1) First, it briefly touches on what is included and excluded from the concept of Humanities; (2) Second, it examines potential issues in terms of the nature and focus of IS research if we allow it to be "enriched" by Humanities. It is done by means of an example of Health Information Systems research in South Africa that draws on the Russian social psychology tradition.

Boundary Issues Related to the Concept of Humanities and its Interaction with Social Psychology and IS Research

Why should there be a contribution from a social psychological perspective as part of this panel? Is social psychology necessarily considered to be part of Humanities? Not according to many definitions—see, for example, the ACM list of terms, where Humanities are J.5. and social psychology is J.4. [ACM, 1998].

Is there anything new in using concepts from social psychology to enrich IS research? Not really, as the link has been visible for decades (see, as an example, the work done by Melone [1990]). Neither is the potential mutual synergy between Humanities and social psychology (interaction of *text* and *context)* considered to be news. This linkage has been thoroughly examined since the 1970s by Geertz [1973; 1983], among others.

Because these linkages are well-established, we would like to turn the issue around, and rather ask the question: Are we as IS researchers brave enough to face up to the challenges posed by Humanities-enriched IS research?—OR—How much enrichment can we endure?

An Example Related to Socio-Technical Health Systems Research in Africa that Draws on Russian Social Psychology

Activity theory had its roots in the work of Vygotsky [1978] and was further developed by various researchers, such as Leont'ev [1978] and others [Bakhurst, 2009]. The developmental work approach, situated within this tradition relates strongly to the concepts developed by Engeström and the work done by CRADLE (in Helsinki) [Engeström, 2000]. The version which found its way into IS research in Africa was methods based on Engeström's work, generically named ActAD [Korpela, 2004]. It consists of a set of symbols, artifacts and procedures, based on AT which has been demonstrated to African audiences. Its adaptation, adoption and use has been extensively promoted, especially in Health Information Systems research [Korpela et al., 1998; Mursu et al., 2003].

Activity as Notion Rooted in "Humanities"

The original conceptualization of activity could be considered almost entirely "humanistic" in the sense that it proposes a "notion" [Bakhurst, 2009] through which human access to reality could be understood. It thus proposes object-oriented activity involving tools (which could be language or signs) as the fundamental way in which humanity appropriates the external world through activity performed on an object.

Activity as Social Psychological Theory

As a further "phase," the "theorizing of activity" already started moving away from this fundamental aspect of AT and theorized the activity aspect of it with the intent to inform practice [Bakhurst, 2009], to the extent that the "third wave" of activity theory is actually again calling for an understanding of the impact of language and culture, as these aspects are considered to have been underdeveloped by the original theorists (see, for example, the arguments by Bødker [1991]).

Activity Informing Socio-Technical IS Research

ActAD had its focus mainly on the development of useful methods for IS practitioners [Korpela et al., 2002]. As such, it mainly constitutes processes [Korpela, 2004], and signs and symbols [Korpela et al., 2008] that could be used in certain phases of ISD. Therefore, it "nods" to its social psychological roots as well as acknowledging "means of communication," yet, it has obviously moved quite some distance from its "Humanities-grounded" origins. ActAD, exported to Africa especially in Health Information Systems research has spawned significant research [Korpela, 2004; Korpela et al., 2002; Mursu et al., 2003]. Significant is the explicit requirement that the method should be appropriated by Africans and should form the basis of indigenized socio-technical IS development processes that contribute to local socio-economic development [De la Harpe et al., 2010]. It relates closely to more generalized views of current "third" generation AT—that account should be taken of, *inter alia*, "voice and identity" [Bakhurst, 2009] that would imply taking stronger account of Humanities-related concepts in language and culture as these aspects of AT have not been developed extensively.

In future, bringing in these Humanities enrichments may result in uncovering the following tension lines related to doing Humanities-enriched IS research in this instance:

a) Language and culture may provide a threat to practitioners and researchers looking for useful methods as both these phenomena tend toward localization and non-generalization.

b) If these concerns prove significant, it may well create significant anomalies and demonstrate AT as inappropriate, leading to new theories and an abandonment of AT.

c) The existing disconnect between the original conceptualization of Vygotsky and the later theorizing of activity and development of methods means that enriching Activity Theory through examination of the effects of language and culture in activity systems will not lead to a return to the fundamental questions related to human nature where AT started. It is, however, not inconceivable that better understanding of the social interaction between humans and tools could lead to insights into fundamental aspects of the human condition.

IX. CONCLUSION

In this article the authors tried to give some recognition for Humanities-enriched IS research that has been done in the past, thus providing evidence of the existence of a sub-discipline. Getting a panel discussion together in itself shows the viability of a guild of members. This new league should eventually become a recognized interest group within Information Systems. Are we as IS researchers brave enough to tackle this challenge and follow the road regardless of where it leads, even if it implies that methods developed on the basis of Humanities enrichment over years may not be standardizable, that we may prove that the social psychological theory that informs the research is inadequate, and that we may end up researching fundamental aspects of the human condition? Where do we choose to draw our boundaries?

ACKNOWLEDGMENTS

An abstract of the original panel discussion was published in the ECIS2010 conference proceedings: Kroeze, J.H. (NWU), Lotriet, H. (UP), Mavetera, N. (NWU), Pfaff, M.S. (Indiana University), Postma, D. (UJ), Sewchurran, K. (UCT) and Topi, H. (Bentley University). 2010. Humanities-enriched information systems. Panel discussion held at ECIS 2010, University of Pretoria, Pretoria, South Africa, 6–9 June 2010. ISBN: 978-0-620-47172-5. Available: http://dspace.nwu.ac.za/handle/10394/3126.

REFERENCES

Editor's Note: The following reference list contains hyperlinks to World Wide Web pages. Readers who have the ability to access the Web directly from their word processor or are reading the article on the Web, can gain direct access to these linked references. Readers are warned, however, that:

1. These links existed as of the date of publication but are not guaranteed to be working thereafter.
2. The contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. The author(s) of the Web pages, not AIS, is (are) responsible for the accuracy of their content.
4. The author(s) of this article, not AIS, is (are) responsible for the accuracy of the URL and version information.

ACM (1998) *The 1998 ACM Computing Classification System—Association for Computing Machinery.* http://www.acm.org/about/class/ (current March 13, 2011).

Allen, G. and S.T. March (2006) "A Critical Assessment of the Bunge-Wand-Weber Ontology of Conceptual Modeling", *Proceedings of the Workshop on Information Technology and Systems (WITS),* Dec., Milwaukee, WI.

Arnold, M. (1960) *On the Classical Tradition: The Complete Prose Works of Matthew Arnold, Vol. I,* Ann Arbor, MI: University of Michigan Press.

Aβmann, U., S. Zschaler, and G. Wagner (2006) "Ontologies, Meta-Models, and the Model Driven Paradigm" in Calero, C., F. Ruiz, and M. Piattini (eds.) *Ontologies for Software Engineering and Software Technology*, Berlin: Springer Verlag, pp. 249–274.

Bakhurst, D. (2009) "Reflections on Activity Theory", *Educational Review* (61), pp. 197–210.

Ben Sta, H. (2010) "Conceptualization of a Framework Based on the Ontology for the Construction [of] an Objects Model", *Proceedings of the 14th IBIMA 2010 Conference: Business Transformation through Innovation and Knowledge Management: An Academic Perspective*, June 23–24, Istanbul, Turkey, pp. 2058–2070.

Bijker, W., T. Hughes, and T. Pinch (eds.) (1987) *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*, Cambridge, MA: MIT Press.

Bloomfield, B. and C. McLean (1996) "Madness and Organization: Informed Management and Empowerment" in Orlikowski, W. et al. (eds.), *Information Technology and Changes in Organization Work,* London, England: Chapman & Hall, pp. 371–392.

Bødker, S. (1991) “Activity Theory as a Challenge to Systems Design” in Nissen, H.E., H.K. Klein, and R. Hirschheim (eds.) *Information Systems Research: Contemporary Approaches and Emergent Traditions*, Amsterdam, The Netherlands: Elsevier.

Bolter, J. and D. Gromala (2006) “Transparency and Reflectivity: Digital Art and the Aesthetics of Interface Design” in Fishwick, P.A. (ed.) *Aesthetic Computing,* Cambridge, MA: MIT Press, pp. 369–382.

Bowen, P., R. O’Farrell, and F. Rohde (2009) “An Empirical Investigation of End-User Query Development: The Effects of Improved Model Expressiveness vs. Complexity”, *Information Systems Research* (20)4, pp. 565–584.

Bowker, G. and S. Star (2000) *Sorting Things Out: Classification and its Consequences*, Cambridge, MA: MIT Press.

Bowker, G., S. Timmermans, and S. Star (1996) “Infrastructure and Organizational Transformation: Classifying Nurses’ Work” in Orlikowski, W. et al. (eds) *Information Technology and Changes in Organization Work,* London, England: Chapman and Hall, pp. 344–370.

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

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

Burton-Jones, A. and P. Meso (2008) “The Effects of Decomposition Quality and Multiple Forms of Information on Novices' Understanding of a Domain from a Conceptual Model”, *Journal of the Association for Information Systems* (9)12, pp. 748–802.

Callon, M. and B. Latour (1981) “Unscrewing the Big Leviathan: How Actors Macro-Structure Reality and How Sociologists Help Them to Do So” in Knorr-Cetina, K. and A. Cicourel (eds.) *Advances in Social Theory and Methodology: Toward an Integration of Micro and Macro-Sociologies,* Boston, MA: Routledge and Kegan Paul, pp. 277–303.

Cassel, L.N. et al. (2007a) “The Computing Ontology—Application in Education”, *ACM SIGCSE Bulletin* (39)4, pp. 171–183.

Cassel, L.N. et al. (2007b) “The Computing Ontology Project: The Computing Education Application” in *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education—SIGCSE '07*, Covington, KY, pp. 519–520.

Chou, C. and Y. Chi (2010) “Developing Ontology-Based EPA for Representing Accounting Principles in a Reusable Knowledge Component”, *Expert Systems with Applications* (37)3, pp. 2316–2323.

Corcho, O., M. Fernandez-Lopez, and M. Gomez-Perez (2006) “Ontological Engineering: Principles, Methods, Tools and Languages” in Calero, C., F. Ruiz, and M Piattini (eds.) *Ontologies for Software Engineering and Software Technology,* New York, NY: Springer Verlag.

Cretu, L.G. (2010) “Business Process Oriented Software Engineering”, *Proceedings of the IBIMA 2010 Conference: Business Transformation through Innovation and Knowledge Management: An Academic Perspective,* June 23–24, Istanbul, Turkey, pp. 1773–1786.

Csikszentmihalyi, M. and I. Csikszentmihalyi (eds.) (1988) *Optimal Experience: Psychological Studies of Flow in Consciousness,* Cambridge, England: Cambridge University Press.

Dahlbom, B. and L. Mathiassen (1993) *Computers in Context: The Philosophy and Practice of Systems Design,* Oxford: NCC Blackwell.

De La Harpe, R. et al. (2010) *Towards a Socio-Technical Approach in ICT-Supported Socio-Economic Development*, Durban, South Africa: IST Africa.

Ding, Y. and F. Foo (2002) “Ontology Research and Development: Part 1—A Review of Ontology Generation”, *Journal of Information Science* (28)2, Thousand Oaks, CA: Sage, pp. 123–136.

Donnelly, J.F. (1999) "Schooling Heidegger: On Being in Teaching”, *Teaching and Teacher Education* (15), pp. 933–949.

Dristas, S. et al. (2005) “Employing Ontologies for the Development of Security Critical Applications: The Secure e-Poll Paradigm.” Proceedings of the International Conference on eBusiness, eCommerce and EGovernment held at Turku. IFIP I3E International Conference on eBusiness,eCommerce, and eGovernment, M. Funabashi, A. Grzech, (Eds.), pp. 187-201, October 2005, Poznan, Poland, Springer

Engeström, Y. (2000) “Activity Theory as a Framework for Analyzing and Redesigning Work”, *Ergonomics* (43), pp. 960–974.

Falbo, R.A., G. Guizzardi, and K.C. Duarte (2002) "An Ontological Approach to Domain Engineering: International Conference of Software Engineering and Knowledge Engineering", SEKE'02, July 15–19, Ischia, Italy.

Fishwick, P.A. (ed.) (2006) *Aesthetic Computing*, Cambridge, MA: MIT Press.

Geertz, C. (1973) *The Interpretation of Cultures: Selected Essays*, New York, NY: Basic Books.

Geertz, C. (1983) *Local Knowledge: Further Essays in Interpretive Anthropology*, New York, NY: Basic Books.

Gemino, A. and Y. Wand (2005) "Complexity and Clarity in Conceptual Modeling: Comparison of Mandatory and Optional Properties", *Data & Knowledge Engineering* (55)3, pp. 301–326.

Ghattas, J. and P. Soffer (2009) "Evaluation of Inter-Organizational Business Process Solutions: A Conceptual Model-Based Approach", *Information Systems Frontiers* (11)3, pp. 273–291.

Gregor, S. (2006) "The Nature of Theory in Information Systems", *MIS Quarterly* (30)3, pp. 612–642.

Grover, V., D. Straub, and P. Galluch (2009) "Turning the Corner: The Influence of Positive Thinking on the Information Systems Field (Editor's Comments)", *MIS Quarterly* (32)1, pp. iii–viii.

Gruber, T. (1995) "Toward Principles for the Design of Ontologies Used for Knowledge Sharing", *International Journal of Human–Computer Studies* (43)5–6, pp. 907–928.

Gruber, T. (2008) "Ontology" in Liu, L. and M. Tamer Ozsu (eds.) *Encyclopedia of Database Systems*, New York, NY: Springer Verlag.

Gruber, T.A. (1993) "Transition Approach to Portable Ontology Specifications", *Knowledge Acquisition* (5) pp. 199–20.

Guarino, N. (1998) "Formal Ontology and Information Systems", *Proceedings of FOIS'98,* June 6–8, Trento, Italy, Amsterdam, Netherlands: IOS Press, pp. 3–15.

Guarino, N., M. Carrara, and P. Giaretta (1994) "Formalising Ontological Commitments", *Proceedings of the* 12th National Conference on Artificial Intelligence *(AAAI-94),* Seattle, Morgan Kaufmann, *pp.* 560-567.

Harris, M.L., A.R. Hevner, and R.W. Collins (2009) "Controls in Flexible Software Development", *Communications of the Association for Information Systems* (24)43, Berkely Electronic Press, pp. 757–776.

Hevner, A.R. et al. (2004) "Design Science in Information Systems Research", *MIS Quarterly* (28), pp. 75–105.

Hofferer, P. (2007) "Achieving Business Process Model Interoperability Using Metamodels and Ontologies" in Khosrow-Pour, M. (ed.) *Managing Worldwide Operations and Communications with Information Technology, Proceedings of 2007 Information Resources Management Association, International Conference,* May 19–23, Vancouver, British Columbia, Canada. Published in book format and on CD by IGI Publishing, Hershey, PA (CD ISBN: 978-159904930-4) (Hardcover ISBN: 978-1-59904-929-8), pp. 1620–1631.

Hohmann, C. (2007) "Emotional Digitization: A Reflexive Examination from the View of the Industry", *International Journal of Technology and Human Interaction* (3)1, pp. 17–29.

Isabella, C.A. (1990) "Evolving Interpretations as a Change Unfolds: How Managers Construe Key Organizational Events", *The Academy of Management Journal* (33)1, pp. 7–41.

Kawalek, P. and J. Leonard (1996) "Evolutionary Software Development to Support Organizational and Business Change: a Case Study Account", *Journal of Information Technology* (11)3, pp. 185–198.

Korpela, M. (2004) "ActAD in Information Systems Analysis and Design: Academic Reflection" in Bertelsen, O.W., M. Korpela, and A. Mursu (eds.) *The First International Workshop on Activity Theory-Based Practical Methods for IT-Design,* Copenhagen, Denmark, University of Aarhus, Sept. 2-3, 2004, DAIMI report, pp. 134-136.

Korpela, M., R. De La Harpe, and I. Luukkonen (2008) "Depicting the Landscape Around Information Flows: Methodological Propositions", *SIG GlobDev's First Annual Workshop*, Dec. 13, Paris, France.

Korpela, M., A. Mursu, and H.A. Soriyan (2002) "Information Systems Development as an Activity", *Computer Supported Cooperative Work* (11), pp. 111–128.

Korpela, M. et al. (1998) "Community Participation in Health Informatics in Africa: An Experiment in Tripartite Partnership in Ile-Ife, Nigeria", *Computer Supported Cooperative Work* (7), pp. 339–358.

Kroeze, J.H. (2010a) "Ontology Goes Postmodern in ICT" in Kotzé, P. et al. (eds.) *Fountains of Computing Research—Proceedings of SAICSIT 2010 (Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists),* Oct. 11–13, Bela Bela, South Africa, CSIR Meraka Institute, A Volume in the ACM International Conference Proceedings Series, ACM Press, ACM ISBN: 978-1-

60558-950-3, pp.153-159, http://portal.acm.org/citation.cfm?id=1899503.1899520&coll=DL&dl=GUIDE&CFID=6036717&CFTOKEN=39458455 (current Jan. 14, 2011). Postprint available: http://dspace.nwu.ac.za/handle/10394/3757 (current Jan. 14, 2011).

Kroeze, J.H. (2010b) "The Mutualistic Relationship between Information Systems and the Humanities" (full paper, edited version of inaugural lecture), *Proceedings of the 15th International Business Information Management Association Conference (15th IBIMA),* Nov. 6–7, Cairo, Egypt, pp. 915–927. (Knowledge Management and Innovation: A Business Competitive Edge Perspective, edited by Khalid S. Soliman. On CD: ISBN: 978-0-9821489-4-5), http://hdl.handle.net/10394/3824 (current Jan. 14, 2011).

Latour, B. (2004) "Why Has Critique Run out of Steam? From Matters of Fact to Matters of Concern", *Critical Inquiry* (30), pp. 225–248.

Law, J. (1986) "On the Methods of Long-Distance Control: Vessels, Navigation and the Portuguese Route to India" in Law, J. (ed.) *Power, Action and Belief,* London, England: Routledge & Kegan Paul, pp. 234–263.

Law, J. (1999) "After ANT: Complexity, Naming and Topology" in Law, J. and J. Hassard (eds.) *Actor Network Theory and After*, Oxford, England: Blackwell, p. 1014.

Lemmens, R. (2006) *Semantic Interoperability*, Ph.D. Dissertation, ITC, Netherlands, http://www.ncg.knaw.nl/publicaties/geodesy/pdf/63lemmens.pdf (current June 17, 2008).

Leont'ev, A. (1978) *Activity, Consciousness, and Personality*, Englewood Cliffs, NJ: Prentice-Hall.

Malina, R.F. (2006) "A Forty-Year Perspective on Aesthetic Computing in the Leonardo Journal" in Fishwick, P.A. (ed.) *Aesthetic Computing,* Cambridge, MA: The MIT Press, pp. 43–52.

Maturana, H.R. and B. Poerksen (2004) *From Being to Doing: The Origins of the Biology of Cognition,* Netherlands: Carl-Auer.

Mavetera, N. (2007) "A Comprehensive Ontology-Driven Software Development Architecture: A Holistic Approach to Developing Romantic Software Products" in Khosrow-Pour, M. (ed.) *Managing Worldwide Operations and Communications with Information Technology, Proceedings of 2007 Information Resources Management Association, International Conference,* May 19–23, Vancouver, British Columbia, Canada, published in book format and on CD by IGI Publishing, Hershey, PA (CD ISBN: 978-159904930-4) (Hardcover ISBN: 978-1-59904-929-8).

Mavetera, N. and J.H. Kroeze (2010) "An Ontology-Driven Software Development Framework", *Proceedings of the 14th International Business Information Management Association Conference (14th IBIMA),* June 23–24, Istanbul, Turkey, pp. 1713–1724. (*Business Transformation through Innovation and Knowledge Management: An Academic Perspective,* edited by Khalid S. Soliman. On CD: ISBN: 978-0-9821489-3-8.)

Melone, N.P. (1990) "A Theoretical Assessment of the User-Satisfaction Construct in Information Systems Research", *Management Science* (36), pp. 76–91.

Merleau-Ponty, M. (1945) *Phenomenology of Perception*, London, England: Routledge.

Merleau-Ponty, M. (1964) *Sense and Non-Sense* (Dreyfus, H.L. and P.A. Dreyfus, trans.), Evanston, IL: Northwestern University Press.

Meso, P. and R. Jain (2006) "Agile Software Development: Adaptive Systems Principles and Best Practices", *Information Systems Management* (23)3, June, pp. 19–30.

Mol, A. (2002) *The Body Multiple: Ontology in Medical Practice,* Durham, NC: Duke University Press.

Monod E. and R.J. Boland (2007) "Special Issue on Philosophy and Epistemology: A 'Peter Pan Syndrome'?" (editorial), *Information Systems Journal,* (17), pp. 133–141.

Mursu, A. et al. (2003) "Identifying Software Project Risks in Nigeria: An International Comparative Study", *European Journal of Information Systems* (12), pp. 182–194.

Neches, R. et al. (1991) "Enabling Technology for Knowledge Sharing", *AI Magazine,* Winter, pp.36–56.

Norman, D.A. (2004) *Emotional Design*, New York, NY: Basic Books.

Parsons, J. and Y. Wand (2008) "Using Cognitive Principles to Guide Classification in Information Systems Modeling", *MIS Quarterly* (32)4), pp. 839-868.

Picard, R.W. (1997) *Affective Computing*, Cambridge, MA: MIT Press.

Quine, W. (1953) "On what there is", in *From a Logical Point of View,* New York, NY: Harper & Row.

Recker, J. et al. (2009) "Business Process Modeling—A Comparative Analysis", *Journal of the Association for Information Systems* (10)4, pp. 333–363.

Rosemann, M. and B. Wyssuek (2005) "Enhancing the Expressiveness of the Bunge-Wand-Weber Ontology", *Proceedings of the Eleventh Americas Conference on Information Systems*, Aug., 11–14, Omaha, NE.

Ruiz, F. and J.R. Hilera (2006) "Using Ontologies in Software Engineering and Technology" in Calero, C., F. Ruiz, and M. Piattini (eds.) *Ontologies for Software Engineering and Software Technology,* New York, NY: Springer Verlag.

Sarantakos, S. (1997) *Social Research, 2nd edition*, New York, NY: Palgrave.

Scott, E. and K. Sewchurran (2008) "Reflection-in-Action: Using Experience to Reconstruct Meaning in a Learning Environment" (CITE2008), International Conference on Information Technology in Education, Dec. 12–14, Wuhan, China: IEEE.

Scott, S. and E. Wagner (2003) "Networks, Negotiations, and New Times: The Implementation of Enterprise Resource Planning into an Academic Administration", *Information and Organization* (13)4, pp. 285–313.

Sewchurran, K. (2008) "Toward an Approach to Create Self Organising and Reflexive Information Systems Project Practitioners", *International Journal of Managing Projects in Business* (1)3, pp. 316–333.

Sewchurran, K. and E. Scott (2009) "Learning and Making Sense of Project Phenomena in Information Systems Education", *The 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human,* Seoul, Korea: Association of Computing Machinery (ACM).

Shanks, G., E. Tansley, and R. Weber (2003) "Representing Composites in Conceptual Modeling", *Communications of the ACM* (47)7, pp. 77–80.

Sheridan, T.B. (1992) "Musings on Telepresence and Virtual Presence", *Presence: Teleoperators and Virtual Environments* (1)1, pp. 120–126.

Shoib, G. and J. Nandhakumar (2009) "Using Social Theory to Make Sense of IS: What's It All About?" (guest editorial), *Information Technology & People* (22)1.

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

Simonov, M., A. Gangemi, and M. Soroldoni (2004) "Ontology-Driven, Natural Language Access to Legacy and Web Services in the Insurance Domain", in Wecel K (ed.), Proceedings of the 7th International Conference on Business Information Systems (BIS), 2004 , pp. 1–10.

Smith, B. (2003) "Ontology" in Floridi, L. (ed.) *Blackwell Guide to the Philosophy of Computing and Information*, Oxford, England: Blackwell, pp. 155-166.

Snow, C. (1959) *The Two Cultures*, Cambridge, England: Cambridge University Press.

Soffer, P. et al. (2001) "Modelling Off-the-Shelf Information Systems Requirements: An Ontological Approach", *Requirements Engineering Journal* (6), pp. 183–199.

Sowa, F.J. (2000) "Ontology, Metadata and Semiotics" in Ganter and Mineau (eds.) *Conceptual Structures in Logical, Linguistic and Computational Issues* (lecture notes in AI #1867), Berlin, Germany: Springer-Verlag, pp. 55–81.

Stamper, R. (1992) "Signs, Organisations, Norms and Information Systems", *Proceedings of the 3rd Australian Conference on Information Systems*, Wollongong, Australia, October 1992 (keynote).

Strasser, S. (1985) *Understanding and Explanation: Basic Ideas Concerning the Humanity of the Human Sciences,* Pittsburgh, PA: Duquesne University Press.

Studer, R., R .Benjamins, and D. Fensel (1998) "Knowledge Engineering: Principles and Methods", *Data and Knowledge Engineering* (25), pp.161–197.

Tarnas, R. (1991) *The Passion of the Western Mind: Understanding the Ideas That Have Shaped Our World View*, London, England: Pimlico.

Tiger, L. (2000) *The Pursuit of Pleasure*, New Brunswick, NJ: Transaction Publishers.

Tractinsky, N. (1997) "Aesthetics and Apparent Usability: Empirically Assessing Cultural and Methodological Issues", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* Atlanta, GA, March 22-27, 1997, pp. 115–122.

Trim, P.R.J. and Y.I. Lee (2004) “A Reflection on Theory Building and the Development of Management Knowledge”, *Journal of Management Decision* (42)3/4, pp. 473–480.

Tufte, E.R. (1997) *Visual Explanations: Images and Quantities, Evidence and Narrative,* Cheshire, CT: Graphics Press.

Vygotsky, L.S. (1978) *Mind in Society,* Cambridge, MA: Harvard University Press.

Wand, Y. and R. Weber (1990) “Toward a Theory of the Deep Structure of Information Systems”, *Proceedings of the Eleventh International Conference of Information Systems,* Copenhagen, Denmark, Dec. 16-19, 1990, pp. 61–71.

Wand, Y. and R. Weber (1995) “On the Deep Structure of Information Systems”, *Journal of Information Systems* (5), pp. 203–223.

Wand, Y. and R. Weber (2002) “Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda”, *Information Systems Research* (13)4, pp. 363–376.

Weber, R. (1997) “Ontological Foundations of Information Systems”, *Coopers & Lybrand Research Methodology* (4).

Weber, R. (2003) “Conceptual Modelling and Ontology: Possibilities and Pitfalls”, *Journal of Database Management* (14)3, pp. 1–20.

Winter, S. and M. Tomko. (2006) “Translating the Web Semantics of Georeferences” in Taniar, D. and J. Wenny Rahayu (eds.) *Web Semantics and Ontology,* Hershey, PA: Idea Group Publishing.

Zúñiga, G.L. (2001) “Ontology: Its Transformation from Philosophy to Information Systems”, *FOIS’01,* Oct., 17–19, Ogunquit, ME .

ABOUT THE AUTHORS

Jan Kroeze has Ph.D.s in Semitic Languages and in Information Technology. He is a professor of Information Systems in the School of Computing at UNISA, South Africa. His current research focuses on formal ontologies and Humanities-enriched IS.

Hugo Lotriet has a Ph.D. in Engineering. He is a professor of Information Systems in the School of Computing at UNISA, South Africa. His current research focuses on socio-technical aspects of the adoption and use of Information Systems in organizations and society.

Nehemiah Mavetera has a Master’s degree in Geoinformation Science and Earth Observation, and he is currently working on the final submission of his doctoral thesis on ontology-driven software development. He is a senior lecturer in Information Systems on the Mafikeng campus of the North-West University, South Africa.

Mark Pfaff has a Ph.D. in Information Sciences and Technology from the Pennsylvania State University. He is an assistant professor of Media Arts and Sciences in the School of Informatics, Indianapolis, IN, US. His research uses games and simulations to study the effects of emotion and stress in technologically complex work environments.

Dirk Postma has a Ph.D. in Informatics. He is a senior lecturer in the Department of Education Studies, University of Johannesburg, South Africa, where he teaches educational theory. His research focus is the philosophy and sociology of technology.

Kosheek Sewchurran has a Ph.D. in Information Systems and he is a senior lecturer at the University of Cape Town, South Africa, in the Department of Information Systems. His research focuses on IS project management, IS graduate attributes and the applications of Systems Thinking.

Heikki Topi has a Ph.D. in Management Information Systems. He is a professor of Computer Information Systems and Associate Dean at Bentley University, Waltham, MA, US. His research focuses on the usability of large-scale enterprise systems, Information Systems curricula, and the organization and meaning of key concepts in computing.

Communications of the Association for Information Systems

ISSN: 1529-3181