1990

TOWARD A THEORY OF THE DEEP STRUCTURE OF INFORMATION SYSTEMS

Yair Wand
The University of British Columbia

Ron Weber
The University of Queensland

Recommended Citation
Wand, Yair and Weber, Ron, "TOWARD A THEORY OF THE DEEP STRUCTURE OF INFORMATION SYSTEMS" (1990). *ICIS 1990 Proceedings*. 3.
http://aisel.aisnet.org/icis1990/3

TOWARD A THEORY OF THE DEEP STRUCTURE OF INFORMATION SYSTEMS

Yair Wand
The University of British Columbia

Ron Weber
The University of Queensland

ABSTRACT

The deep structure of an information system comprises those properties that manifest the meaning of the real-world system that the information system is intended to model. In this paper we describe three models that we have developed of information systems deep-structure properties. The first, the representational model, proposes a set of constructs that enable the ontological completeness of an information systems grammar to be evaluated. The second, the state-tracking model, proposes four requirements that information systems must satisfy if they are to faithfully track the real-world system they are intended to model. The third, the good-decomposition model, proposes a set of necessary conditions that an information system must meet if it is to be well decomposed. The three models facilitate the evaluation of grammars used to analyze, design, and implement information systems and specific scripts that represent implemented information systems.

1. INTRODUCTION

Over the last few years, we have been attempting to build formal models of information systems. Our purposes are twofold. First, we seek to understand and predict certain aspects of the structure and behavior of "good" information systems. In particular, we are focusing on those properties that an information system must possess if it is to manifest the *meaning* of the real-world system it is intended to model. Second, we seek to understand and predict the characteristics of "good" grammars that can be used to describe information systems and the real-world systems they model.

In this paper we provide an overview and synthesis of our work. First we articulate a particular view of information systems that forms the basis of and motivates the nature of the formal models we have developed. Next we describe the set of major premises that underlie our formal models. We then provide a brief description of the formal models and seek to show their potential power by using them to evaluate a systems analysis and design tool. Finally, we discuss some future research directions and present some brief conclusions.

2. OUR VIEW OF AN INFORMATION SYSTEM

Our formal models of an information system are motivated by a particular *view* or conception of information systems that we adopt. Specifically:

> *We conceive of an information system as an object that can be studied in its own right, independently of the way it is deployed in its organizational and social context and the technology used to implement it.*

In other words, when modeling an information system, we are not concerned with the way it is managed in organizations, the characteristics of its users, the way it is implemented, the way it is used, the impact it has on such factors as quality of working life or the distribution of power in organizations, or the type of hardware or software used to make it operational.[1] Instead, we are concerned only with information systems as independent artifacts that bear certain relationships to the real-world system they are intended to model. This view is not intended to denigrate the importance of deployment and technology issues to the successful development, implementation, and use of information systems. Rather, we seek to show that advantages accrue from decoupling the study of these issues from the study of certain other properties that can be identified when information systems are conceived as independent artifacts (Weber 1987).

Next we distinguish between three sets of characteristics of the information systems object. The first set comprises the "*surface-structure*" characteristics of the information system. These characteristics manifest the nature of the interface between the information system and its users and organizational environment. For example, the type of interactive dialog used in the system or the format of reports produced by the system are surface-structure characteristics. The second set comprises the "*deep-structure*" characteristics of the information system. These characteristics manifest the *meaning* of the real-world system that the information system is intended to model.

For example, the rules embodied in an accounting system that indicate how transactions are to be posted to ledgers are deep-structure characteristics.[2] The third set comprises the *"physical-structure"* characteristics of the information system. These characteristics manifest the technology used to implement the system. For example, the way in which data in the system is assigned to a mass-storage device or the communications protocol chosen for message transmission in the system are physical-structure characteristics.

In our formal models, *we focus only on the deep-structure characteristics of an information system.* We choose this stance because we seek to provide inherent stability to the models we develop. We contend that the surface-structure and physical-structure properties of an information system inevitably follow the whims of changing social circumstances and changing technology. The deep-structure properties, on the other hand, tend to be more robust to change. Moreover, the surface-structure and physical-structure properties of an information system can be changed without changing its deep structure. For example, the user interface may be modified or the system implemented on a new machine with no effect on the meaning of the information processing carried out by the system (Benbasat and Wand 1984; Linton, Vlissides, and Calder 1989). In this respect we seek to develop models that lie at the core of information system design.[3]

Our focus on deep-structure properties, however, results in only a limited notion of "goodness" in an information system. Specifically, given our view, we assess goodness in terms of how well information systems embody the meaning of the real-world system they are intended to model. Clearly, this notion of goodness is limited. It ignores the significant impact that surface-structure and physical-structure properties might have on an information system's effectiveness and efficiency.[4]

3. THE UNDERLYING PREMISES

Our formal models are motivated by four premises that reflect our view of an information system and its relationship to the real-world system it is intended to model. We begin with the first premise, which we call the *fundamental* premise because it underlies all our work:

> ***The Fundamental Premise***: A physical-symbol system has the necessary and sufficient properties to represent real-world meaning.

Note, this premise is an adaptation of Newell and Simon's (1976) physical-symbol system hypothesis. Whereas Newell and Simon hypothesize a physical-symbol system has the necessary and sufficient properties *for intelligent action*, we adopt a weaker hypothesis relating only to real-world meaning.

Next we give three *working* premises that motivate different formal models we have developed. The first relates to our conception of information systems as representations of the real world:

> ***Working Premise 1***: An information system is an artifactual *representation* of a real-world system as perceived by someone, built to perform information processing functions.

The representation premise reflects our belief that information systems are primarily intended to model the states and behavior of some existing or conceived real-world system. This premise has motivated our development of formal models that help identify those deep-structure properties an information system must possess if it is to be a good representation of the real-world system it is intended to model.

The second working premise relates to our conception of information systems as artifacts intended to track the behavior of real-world systems:

> ***Working Premise 2***: An information system is a *state-tracking mechanism* for the real-world system it is intended to model.

The state-tracking premise reflects our belief that information systems are tools constructed by humans to reduce the financial or cognitive costs of monitoring some real-world system. The real-world system may have a physical manifestation; for example, it may be a working transaction processing system. Alternatively, it may be a conceptual real-world system; for example, it may be a decision support system that simulates some world that exists only in the mind of the user. When the real-world system changes states, the information system should change states accordingly. The state-tracking premise has motivated our development of formal models that help identify those deep-structure properties an information system must possess if it is to faithfully track the behavior of the real-world system it is intended to model.

The third working premise relates to our conception of the way deep-structure properties must be organized in good information systems:

> ***Working Premise 3***: A good information system is *well decomposed.*

The good-decomposition premise reflects our belief that (a) the behavior of information systems having well-decomposed deep structures is easier to understand and predict, and (b) in some sense these systems are more effective and efficient. The importance of good decompositions in general has widespread acceptance in both the computer science and information systems disciplines (Gane and Sarson 1979; Yourdon and Constantine 1979). Furthermore, substantial psychological research supports the notion that human information processing performance depends upon how well semantic memory is structured

(Ashcraft 1989). The good-decomposition premise has motivated our development of formal models to improve our understanding of the meaning of good decompositions and to identify those characteristics information systems must possess if their deep structures are to be well decomposed.

4. THE FORMAL MODELS

In this section we describe three formal models we have developed based upon our underlying premises. We provide only brief, intuitive explanations of the models. More rigorous expositions are available elsewhere (Wand and Weber 1988, 1989c, 1990, 1991).

4.1 The Representational Model

Design and implementation of information systems is an iterative process (Figure 1). During each iteration, a script is generated using some type of grammar that describes the structure and behavior of a real-world system. The grammars used to generate the early scripts employ human-oriented symbols. Later scripts are generated using machine-oriented grammars. Each script is progressively transformed into a new script until one is generated that can be read, interpreted, and executed by a machine.

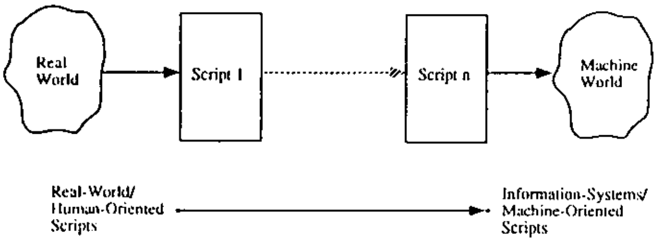

Figure 1. A Transformational Model of Information Systems Analysis, Design, and Implementation

If an information system is intended to be a representation of a real-world system, the grammars used during the design and implementation process must be capable of fully describing the structure (statics) and behavior (dynamics) of the real world. As one script is transformed into another script, the characteristics of the real-world system of interest must be preserved. Even the final script that is read, interpreted, and executed by a machine must still preserve these real-world characteristics. In short, the salient real-world system characteristics must be carried across scripts as *invariants* in the transformation process.

What are these real-world characteristics that must be preserved? To obtain an answer to this question, we have turned to the discipline of philosophy. Within philosophy, the structure and behavior of the real world are the concern of ontologists. Accordingly, we have sought ontological models that would enable us to identify those deep-structure properties an information system must possess if it is to be a good representation of the real-world system it is intended to model. We have chosen and extended an ontological formalism developed by Bunge (1977, 1979) to address information-systems representational issues. Bunge's ontology attracted us because many concepts he examines are directly applicable to the information-systems and computer-science domains. Subsequently we have found his model robust under extensions we have made to include various phenomena not incorporated within the original formalism.

The purpose of the ontological model we have proposed is to define a set of constructs that are necessary and sufficient to describe the structure and behavior of the real world.[5] If this set of constructs can be identified, they provide a benchmark to evaluate whether those grammars used to describe real-world systems are *ontologically complete*. If a grammar cannot represent some type of ontological construct, we predict that descriptions of the real-world system generated using this grammar will be deficient. The nature of the missing ontological construct may provide insights into the likely deficiencies of the scripts generated using the grammar. To draw an analogy, the ontological model fulfills the same purpose as the relational calculus in relational database management theory. Recall, the relational calculus enables relational languages to be evaluated to determine whether they are relationally complete (Codd 1972).[8]

Table 1 provides an overview of the constructs we have proposed so far in our ontological models. Currently we claim these constructs are *necessary* components that must be captured in a grammar used to generate either good human-oriented or good machine-oriented descriptions of real-world systems. Whether they are sufficient constructs is an ongoing research issue.

4.2 The State-Tracking Model

At each stage in the analysis, design, and implementation of an information system, the structure and behavior of the information system must be determined from the script(s) used to describe it. In the context of our state-tracking premise, we are concerned with whether the scripts reveal that the information system will faithfully track the real-world system it is intended to model. To the extent that the scripts are incomplete (that is, they reveal the information system cannot faithfully track the real-world system), additional knowledge must be provided by the processing mechanism that interprets the scripts.

On the basis of the state-tracking models we have constructed, we conclude that four conditions are *necessary and sufficient* conditions for an information system to faithfully track the real-world system it is intended to model (Wand and Weber 1988, 1990). The first requirement that must hold is the *mapping requirement*. It is

Table 1. Ontological Constructs in Our Representational Model

Thing	A thing is the elementary unit in our ontological model. The real world is made up of things. A composite thing may be made up of other composite things or primitive things.
Properties	Things are known via their properties. A property maps the thing into some value. A property of a composite thing that belongs to a component thing is called an hereditary property. A property that does not belong to any of the composing things is called an emergent property.
State	The vector of values for all properties of a thing is the state of the thing.
Conceivable State Space	The set of all states that the thing might ever assume is the conceivable state space of the thing.
State Law	A state law restricts the values of the properties of a thing to a subset that is deemed lawful because of natural laws or human laws.
Lawful State Space	The lawful state space is the set of states of a thing that comply with the state laws of the thing. The lawful state space is usually a proper subset of the conceivable state space.
Event	An event in a thing is a change of state.
Event Space	The event space of a thing is the set of all possible events that can occur in the thing.
Transition Law	A transition law defines which events in a thing are lawful.
Lawful Event Space	The lawful event space is the set of all events in a thing that are lawful.
History	The chronologically-ordered states that a thing traverses in time are the history of the thing.
Coupling	A thing acts on another thing if its existence affects the history of the other thing. The two things are said to be coupled or interact.
System	A set of things is a system if, for any bi-partitioning of the set, couplings exist among things in the two subsets.
System Composition	The things in the system are its composition.
System Environment	Things that are not in the system but interact with things in the system are called the environment of the system.
System Structure	The set of couplings that exist among things in the system and things in the system and things in the environment of the system is called the system structure.
Subsystem	A subsystem is a system whose composition and structure are subsets of the composition and structure of another system and whose environment is a subset of the environment of the other system in union with the things that are in the composition of the other system but not in the composition of the subsystem.
System Decomposition	A decomposition of a system is a set of subsystems such that every component in the system is either one of the subsystems in the decomposition or is included in the composition of one of the subsystems in the decomposition.
Level Structure	A level structure defines a partial order over the subsystems in a decomposition to show which subsystems are components of other subsystems or the system itself.
External Event	An external event is an event that arises in a thing, subsystem, or system by virtue of the action of some thing in the environment on the thing, subsystem, or system.
Stable State	A stable state is a state in which a thing, subsystem, or system will remain unless forced to change by virtue of the action of a thing in the environment (an external event).
Unstable State	An unstable state is a state that will be transformed into another unstable state or a stable state by virtue of the action of transition laws.
Internal Event	An internal event is an event that arises in a thing, subsystem, or system by virtue of transition laws in the thing, subsystem, or system.
Well-Defined Event	A well-defined event is an event in which the subsequent state can always be predicted given that the prior state is known.
Poorly-Defined Event	A poorly-defined event is an event in which the subsequent state cannot be predicted given that the prior state is known.

based on our representational premise and ontological formalism and relates to the *structure* of both the real-world and information systems:

> ***Requirement 1***: A one-to-many mapping must exist from the set of real-world system states *into* the set of information system states.

If the mapping requirement is satisfied, at least one information system state exists for every real-world system state. Note, more than one information system state might exist for each real-world system state because of the way the information system is implemented. For example, the system may delay processing of transactions to improve update efficiency. One real-world state may correspond to multiple system states to reflect the variable length of transaction queues that await processing.

The second requirement is the *tracking requirement*. It stipulates that the information system must replicate real-world system behavior:

> ***Requirement 2***: When the real-world system changes states, the information system must be able to change from a state that corresponds to the initial real-world system state to a state that corresponds to the subsequent real-world system state.

Note, the tracking requirement simply says that the transition laws in the information system ensure the information system changes states in a manner corresponding to the state changes occurring in the real-world system. In other words, if independent observers detected a change of state in the information system, they could tell the new state in the real-world system without having to examine it. This requirement implies, therefore, that a homomorphism exists between state transitions in the real-world system and state transitions in the information system.

The mapping and tracking requirements are still insufficient, however, to guarantee that the information system will faithfully represent the real-world system behavior. The first problem may be that relevant events in the real-world system are not reported to the information system. Accordingly, we have the *reporting requirement*, which pertains to external events – events in a system that reflect the influence of the environment:

> ***Requirement 3***: If an external (input) event occurs in the real-world system, an external (input) event that is a faithful representation of the real-world external event must occur in the information system.

This requirement can only be satisfied if an external event occurs in the information system each time an external event occurs in the real-world system. Moreover, the information-system external event must be an accurate and complete representation of the real-world system external event.

The second problem is that the external events occurring in the information system may not arise in the same sequence as the external events occurring in the real-world system. Thus, we have the *sequencing requirement*:

> ***Requirement 4***: The order in which external events occur in the information system must be the same as the order in which external events represented by these information-system external events occur in the real-world system.

The purpose of the sequencing requirement, therefore, is to ensure that the information system does not lose track of the real-world system states because external events are not occurring in the information system in the correct order.

The four state-tracking requirements allow us to carry out two types of evaluations. First, any grammar used to describe an information system can be examined to determine whether it contains components that enable the four requirements to be satisfied. If the grammar does not provide these components, our model predicts that scripts generated using the grammar will be defective. Second, a *particular* script generated via a grammar to describe an information system can be evaluated to determine whether it satisfies the four requirements. A grammar may contain components that enable scripts to be generated which satisfy the four requirements; however, a particular script produced using the grammar still may not satisfy the four requirements.

4.5 The Decomposition Model

A decomposition of a system reflects how it has been broken up into subsystems. The subsystems are usually arranged as a level structure to show how lower-level subsystems are components of higher-level subsystems. From an analysis and design viewpoint, "good" decompositions allow individuals who study and design the system to focus on certain parts of the system somewhat independently of its other parts. Thus, they have a technique for dealing with complexity (Courtois 1985). From an operational and maintenance viewpoint, well-decomposed systems appear to operate more efficiently and are more robust to change (Yourdon and Constantine 1979).

Our decomposition model has enabled us to define the notion of a decomposition precisely and to identify certain characteristics of a good decomposition (Wand and Weber 1989c, 1991). The primary static concepts used in the model are things, couplings between things, subsystems, systems, and level structures (Table 1). The primary dynamic concepts used are stable and unstable states, external events, and internal events (events that reflect the system is attempting to restore itself to a stable state after it has been forced into an unstable state by an external event) (Table 1).

To illustrate the nature of the model, consider an external (input) event that occurs in a subsystem of a system. This external event may reflect the direct influence of the system's environment because the environmental component directly changes one or more of the subsystem's components (Figure 2a). Alternatively, it may reflect the indirect influence of the system's environment because changes in components of other subsystems affect one or more of the subsystem's components. In other words, the subsystem external event arises because the effects of a system-level external propagate through subsystems (Figure 2b).

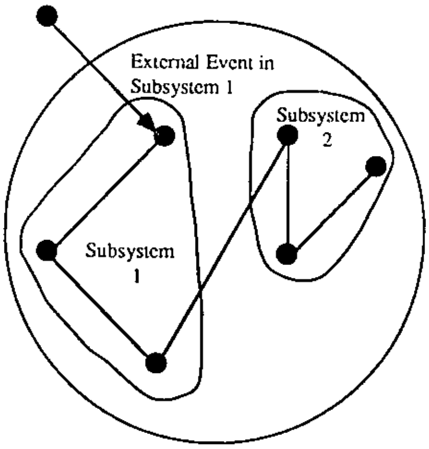

Figure 2(a). Direct Effect of External Event on Subsystem

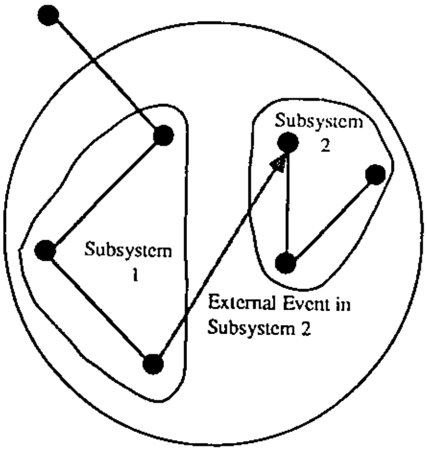

Figure 2(b). Indirect Effect of External Event on Subsystem

If the external event transforms the subsystem to an unstable state, transition laws will act to restore the subsystem to a stable state. The action of transition laws will be manifested as one or more internal events in the subsystem. These internal events may be well-defined in the sense that the subsequent state of the subsystem can be predicted given knowledge of the prior state (Figure 3a). Alternatively, they may be poorly-defined in the sense that the subsequent state of the subsystem can not be predicted given knowledge of the prior state (Figure 3b).

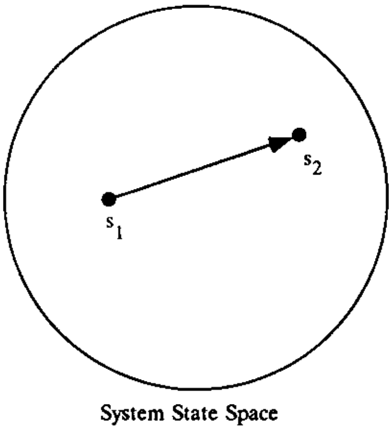

Figure 3(a). Well-Defined Event

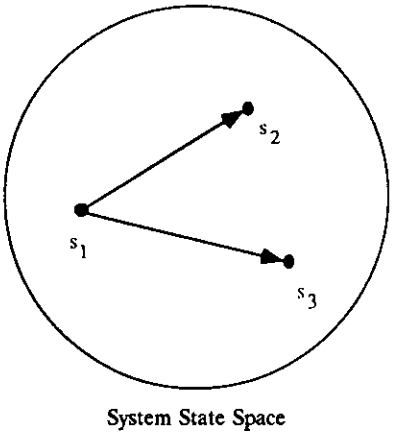

Figure 3(b). Poorly-Defined Event

These notions allow us to define the *characteristics of a good decomposition*:

> ***The Good-Decomposition Proposition***: For a given set of external events at the system level, a decomposition is good *only if* for every subsystem at every level in the level structure of the system an event is either (a) a specified external event or (b) a well-defined internal event.

Three aspects of this proposition are important. First, note that a decomposition is good or poor *only with respect to a certain set of external events at the system level*. This set must be defined at the outset. If it changes, a decomposition may no longer be good when the new external events are considered.

Second, events must be either specified external events or well-defined internal events (Figure 4a). The nature of external events is such that they are often not well defined. In other words, given the system is in a particular state, the new state that arises as a result of an external event cannot always be predicted. For example, the amount of inventory received in an inventory system may depend upon the amount a vendor can provide. Given the current state of inventory, the new state is difficult to predict if the vendor can supply variable amounts. All external events must be specified, however, in a good decomposition. Any poorly-defined event that occurs which is not a specified external event indicates the system has not been well decomposed (Figure 4b).

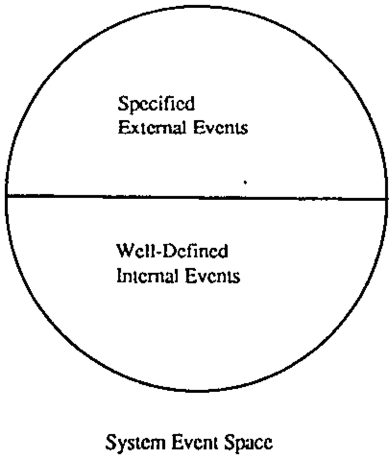

Figure 4(a). System Event Space Under a Good Decomposition

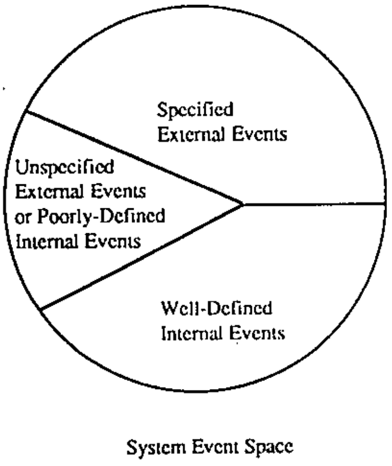

Figure 4(b). System Event Space Under a Poor Decomposition

Third, the proposition states only *necessary* conditions for a good decomposition. Whether they are sufficient conditions is an ongoing research issue. At this stage, however, more than one decomposition of a system may exist that fulfills the good-decomposition condition with respect to a set of external events.

The good-decomposition proposition allows us to establish criteria that can be employed to evaluate both the grammars used to describe systems and the specific scripts generated using these grammars. In the case of grammars, they should contain components that enable their users to design level structures where the event space of each subsystem in each level is partitioned into specified external events and well-defined internal events. In the case of a specific script, all valid interpretations of the script should show that the events occurring in the system are either specified external events or well-defined internal events.

5. AN APPLICATION OF THE FORMAL MODELS

In this section we attempt to show the power of our models by using them to evaluate a widely-used grammar that facilitates undertaking information systems analysis, design, and implementation – namely, the entity-relationship model (ERM).[7] We provide only a brief evaluation of the ERM grammar. A more complete analysis is available elsewhere (Wand and Weber 1989b). Furthermore, since our primary focus is to show how our models can be used, we evaluate the ERM proposed by Chen (1976) rather than the extended ERM (Teorey, Yang, and Fry 1986).

Scripts generated using the ERM grammar are close to the real-world end of the continuum shown in Figure 1. Recall, the ERM is intended to allow "semantic modeling" of the domain of discourse. Supposedly it enables designers and users of an information system to obtain a better understanding of the real-world system that underlies the information system they are intending to build. The mapping between ontological constructs and ERM constructs, therefore, should be fairly direct.

Table 2 shows our evaluation of the ERM to determine whether it is ontologically complete. The table indicates the ERM is deficient primarily in *four* respects:

1. The ERM cannot fully represent states, state spaces, and state laws. Thus, important semantic information about the states a real-world system may traverse and which of these states are lawful may not be captured in the information system.

2. The ERM cannot fully represent events, event spaces, transition laws, and lawful event spaces. Information systems designers must somehow capture the dynamics of the real-world system to be incorporated in the information system using other means.

3. The ERM cannot fully represent the history of a thing, subsystem, or system. A relationship implies that the history of at least one of the entities in the relationship is conditional on the history of the other entity in the relationship. The details of the histories of things, however, are not shown.

Table 2. Evaluation of the Entity-Relationship Model for Ontological Completeness

Thing	Things are represented in the ERM via entities and in some cases relationships. Relationships represent composite things when they are *information-bearing relationships*; that is, when they possess attributes other than the identifiers of the entities that participate in the relationship.
Property	Properties of things are represented in the ERM via attributes and in some cases relationships. If the attributes of a relationship comprise only the identifiers of the entities that make up the relationship (that is, it is a *non-information-bearing relationship*), then the relationship simply manifests properties of the individual entities that make up the relationship. It is not a substantial thing itself.
State	The values of the attributes of entities and relationships at different points in time denote states. However, these values are not represented directly in an ERD. They are provided via supplemental information, e.g., a data dictionary.
Conceivable State Space	The conceivable state space is not represented directly in an ERD. It must be determined from supplemental information, e.g., a data dictionary.
State Law	Only a small amount of state law information is represented in an ERD via referential and cardinality constraints. Other state laws must be determined from supplemental information, e.g., a data dictionary.
Lawful State Space	The lawful state space must be determined from supplemental information, e.g., a data dictionary.
Event	There is no construct in the ERM to represent events.
Event Space	Since there is no construct in the ERM to represent events, the event space also cannot be represented.
Transition Law	There is no construct in the ERM to represent transition laws.
Lawful Event Space	Since events and transition laws cannot be represented in an ERD, the lawful event space also cannot be represented.
History	There is no construct in the ERM to represent history.
Coupling	Some couplings are shown in an ERD via relationships. A relationship means that the history of one entity depends upon the other entity.
System	Providing all couplings between entities in an ERD are shown via relationships, the ERD represents a system.
System Composition	The entities and relationships in an ERD constitute the composition of the system.
System Environment	An ERD may show some entities or relationships that are part of the environment. However, it does not show which entities and relationships are in the composition of the system and which are in the environment of the system unless a boundary is drawn around the entities that are in the composition of the system.
System Structure	An ERD shows the system structure providing (a) all entities in the environment and composition of the system are shown, (b) all couplings between entities in the environment of the system and entities in the composition of the system are shown via relationships, and (c) all couplings between entities in the composition of the system are shown via relationships.
Subsystem	There are no formal constructs for representing a subsystem in an ERM. However, subsystems can be designated by drawing a boundary around entities in an ERD that are coupled (as manifested by relationships) in such a way that the definition of a subsystem is satisfied.
System Decomposition	There are no formal constructs for representing a decomposition in the ERM. However, since a subsystem can easily be designated by drawing a boundary around entities which themselves constitute a system (see above), a decomposition can also be represented by ensuring that a sufficient number of subsystems are designated in this way that they satisfy the definition of a decomposition.
Level Structure	There are no formal constructs for representing a level structure in the ERM. Given that subsystems and a decomposition can be represented by drawing appropriate boundaries around entities, however, a level structure can also be represented by iteratively drawing boundaries around subsystems in such a way that the definition of a level structure is satisfied.
External Event	Since there are no constructs for showing events in the ERM, external events cannot be represented.
Stable State	There are no constructs in an ERM that show which states of a thing, subsystem, or system are stable.
Unstable State	There are no constructs in the ERM that show which states of a thing, subsystem, or system are unstable.
Internal Event	Since there are no constructs for showing events in the ERM, internal events cannot be represented.
Well-Defined Event	Since there are no constructs for representing events in the ERM, well-defined events cannot be represented.
Poorly-Defined Event	Since there are no constructs for representing events in the ERM, poorly-defined events cannot be represented.

4. The ERM does not distinguish stable from unstable states. Furthermore, since it does not represent events, it provides no means of distinguishing between external and internal events and well-defined and poorly-defined events.

In the context of our state-tracking model, the ERM fails to provide constructs that ensure the designer builds information systems that faithfully track the real-world systems they are intended to model. Consider how well a designer who uses the ERM could achieve the four requirements that must be met if an information system is to be a faithful state-tracking mechanism:

1. The mapping requirement may not be satisfied because the ERM is ontologically deficient (see Table 2). In particular, states are not fully represented in the ERM.

2. The tracking requirement may not be satisfied because the ERM does not provide constructs that represent transition laws. Thus, a change of state in the real-world system may not be mirrored by a change of state in the information system.

3. The reporting requirement may not be satisfied because the ERM does not provide constructs to represent events. Thus, the information system may not know that the real-world system has been subject to an external event.

4. The sequencing requirement may not be satisfied because the ERM does not provide constructs to represent events.

In the context of our good-decomposition model, the ERM is again deficient because it is unable to represent events. Thus, external events cannot be distinguished from internal events and well-defined events cannot be distinguished from poorly-defined events. These constructs are needed if the goodness of a decomposition is to be evaluated.

6. FUTURE RESEARCH DIRECTIONS AND CONCLUSIONS

Our current research is aimed at refining and extending our three models. With our representational model, we are pursuing two research directions. First, we are undertaking theoretical research to determine whether new ontological constructs are needed to model real-world systems. Second, we are pursuing empirical work that tests our predictions about the ontological strengths and weaknesses of different information systems grammars.

With our state-tracking model, we are seeking to evaluate information systems grammars to determine whether they provide constructs that enable the four state-tracking requirements to be satisfied. These theoretical evaluations will form the basis of empirical work that tests the relative effectiveness of grammars. In addition, we intend to evaluate implemented information systems to determine how well they ensure the four requirements are satisfied.

With our decomposition model, we are seeking to identify further characteristics of good decompositions. Our ultimate goal is to identify the necessary and sufficient conditions for a good decomposition. In addition, we are evaluating existing information systems analysis, design, and implementation grammars to determine whether they provide constructs that ensure the scripts they generate can lead to good decompositions.

On the basis of our work so far, we believe the ontological approach to understanding and formalizing information systems concepts provides us with the rudiments of a theory of the deep structure of an information system. As a number of writers have observed (e.g., Bubenko 1986), lack of suitable theory has seriously undermined research in the information systems analysis, design, and implementation areas. While our models cannot address all phenomena of interest in these areas, we believe they will prove fruitful in addressing issues concerned with the semantics of information systems.

7. ACKNOWLEDGEMENTS

We are indebted to Frank Land, Michael Lawrence, and Carson Woo for helpful comments on this paper. The research described in this paper was supported in part by an NSERC operating grant and a grant from GWA Ltd.

8. REFERENCES

Ashcraft, M. H. *Human Memory and Cognition*. Glenview, Illinois: Scott, Foresman and Company, 1989.

Benbasat, I., and Wand, Y. "A Structured Approach to Designing Human-Computer Dialogue." *International Journal of Man-Machine Studies,* Volume 21, 1984, pp. 105-126.

Bubenko, J. A., Jr. "Information System Methodologies – A Research Review." In T. W. Olle, H. G. Sol and A. A. Verrijn-Stuart (eds.), *Information Systems Design Methodologies: Improving the Practice*, Amsterdam: North-Holland, 1986, pp. 289-318.

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

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

Chen, P. P. S. "The Entity-Relationship Model – Toward a Unified View of Data." *ACM Transactions on Database Systems*, Volume 1, March 1976, pp. 9-36.

Chomsky, N. *Aspects of a Theory of Syntax.* Cambridge, Massachusetts: Harvard University Press, 1965.

Codd, E. F. "Relational Completeness of Data Base Sublanguages." In R. Rustin (ed.), *Data Base Systems*, Englewood Cliffs, New Jersey: Prentice-Hall, 1972, pp. 65-98.

Courtois, P. J. "On Time and Space Decomposition of Complex Structures." *Communications of the ACM*, Volume 28, June 1985, pp. 590-603.

Culnan, M. "The Intellectual Structure of Management Information Systems, 1972-1982: A Co-citation Analysis." *Management Science*, Volume 32, February 1986, pp. 156-172.

Gane, C., and Sarson, T. *Structured Systems Analysis: Tools and Techniques.* Englewood Cliffs, New Jersey: Prentice-Hall, 1979.

Hirschheim, R., and Klein, H. K. "Four Paradigms of Information Systems Development." *Communications of the ACM*, Volume 10, October 1989, pp. 1199-1216.

Hull, R., and King, R. "Semantic Database Modelling: Survey, Applications, and Research Issues." *ACM Computing Surveys*, Volume 19, September 1987, pp. 201-260.

Linton, M. A.; Vlissides, J. M.; and Calder, P. R. "Composing User Interfaces with InterViews." *IEEE Computer*, Volume 22, February 1989, pp. 8-22.

Newell, A., and Simon, H. A. "Computer Science as Empirical Inquiry: Symbols and Search." *Communications of the ACM*, Volume 19, March 1976, pp. 113-126.

Podger, D. N. "High Level Languages – A Basis for Participative Design." In N. Szyperski and E. Grochla (eds.), *Design and Implementation of Computer-Based Information Systems*, Alphen aan den Rijn, The Netherlands: Sijthoff & Noordhoff, 1979, pp. 243-259.

Teorey, T. J.; Yang, D.; and Fry, J. P. "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model." *ACM Computing Surveys*, Volume 18, June 1986, pp. 197-222.

Wand, Y. "A Proposal for a Formal Model of Objects." In W. Kim and F. Lochovsky (eds.), *Object-Oriented Concepts, Applications, and Databases*, Reading, Massachusetts: Addison-Wesley, 1989, pp. 537-559.

Wand, Y., and Weber, R. "An Ontological Analysis of Some Fundamental Information Systems Concepts." In J. I. DeGross and M. H. Olson (eds.), *Proceedings of the Ninth International Conference on Information Systems*, Minneapolis, Minnesota, December 1988, pp. 213-225.

Wand, Y., and Weber, R. "A Model of Control and Audit Procedure Change in Evolving Data Processing Systems." *The Accounting Review*, Volume 64, January 1989a, pp. 87-107.

Wand, Y., and Weber, R. "An Ontological Analysis of Some Systems Analysis and Design Methods." In E. Falkenberg and P. Lindgreen (eds.), *Information Systems Concepts - An In-Depth Analysis*, Amsterdam: North-Holland, 1989b, pp. 79-107.

Wand, Y., and Weber, R. "A Model of Systems Decomposition." In J. I. DeGross, J. C. Henderson, and B. R. Konsynski (eds.), *Proceedings of the Tenth International Conference on Information Systems*, Boston, Massachusetts, 1989c, pp. 41-51.

Wand, Y., and Weber, R. "Bunge's Ontology as a Formal Foundation for Information Systems Concepts." In P. Weingartner and G. Dorn (eds.), *Studies on Bunge's Treatise*, publication forthcoming, 1990.

Wand, Y., and Weber, R. "An Ontological Model of An Information System." *IEEE Transactions on Software Engineering*, publication forthcoming, 1991.

Waters, S. J. "Towards Comprehensive Specifications." *The Computer Journal*, Volume 22, August 1979, pp. 195-199.

Weber, R. "Toward a Theory of Artifacts: A Paradigmatic Basis for Information Systems Research." *Journal of Information Systems*, Volume 2, Spring 1987, pp. 3-19.

Weber, R. *EDP Auditing: Conceptual Foundations and Practice*, Second Edition. New York: McGraw-Hill, 1988.

Yourdon, E., and Constantine, L. L. *Structured Design: Fundamentals of a Discipline of Computer Program and System Design.* Englewood Cliffs, New Jersey: Prentice-Hall, 1979.

9. ENDNOTES

1. Of course, if these factors are properties of the real world to be modeled in the information system, then the information system must have a representation of them.

2. The distinction between the surface structure and deep structure of an information system is motivated by Chomsky's (1965) distinction between the surface structure (syntax) and deep structure (semantics) in

human language. See also Podger (1979), who describes information systems in terms of concentric domains with varying levels of inertia.

3. In our view, most research in the information systems discipline has focused on management and deployment issues and the surface-structure properties of information systems (Culnan 1986) and most research in the computer science discipline has focused on the physical-structure properties of information systems. The major exceptions have been research on information systems methodologies (Bubenko 1986) and database semantic modeling (Hull and King 1987).

4. We recognize the problems involved in defining information system effectiveness (Weber 1988).

5. More precisely, the ontological constructs deal with someone's perception of the real world. The problematical nature of real-world perceptions in information systems analysis, design, and implementation is well recognized (Hirschheim and Klein 1989).

6. In a similar vein, Waters (1979) has also tried to identify those facts that a specification language must be able to describe for it to be complete.

7. Elsewhere we have used the models to predict the impact of changes to a system on controls and audit procedures (Wand and Weber 1989a) and to better understand the nature of object-oriented design (Wand 1989).