
Hong Zhang, Missouri State University, USA
Rajiv Kishore, State University of New York at Buffalo, USA
Ram Ramesh, State University of New York at Buffalo, USA

ABSTRACT

A conceptual modeling grammar should be based on the theory of ontology and possess clear ontological semantics to represent problem domain knowledge in a precise and consistent manner. In this paper, we follow the notion of ontological expressiveness and conduct an ontological analysis of a newly-developed conceptual modeling grammar termed MibML (Multiagent-based Integrative Business Modeling Language). The grammar is developed to respond to the emerging needs for a special-purpose conceptual modeling grammar for the MIBIS (multi-agent-based integrative business information systems) universe. We assign ontological semantics to the MibML constructs and their relationship using the BWW (Bunge-Wand-Weber) model. This article provides a starting point to further develop ontological principles and step-by-step guidelines to ensure the straightforward mapping from domain knowledge into MibML modeling constructs.

Keywords: Bunge-Wand-Weber (BWW) model; conceptual modeling; MibML

INTRODUCTION

Conceptual modeling is the activity of formally describing some aspects of the physical and social world around us for purposes of understanding and communication (Mylopoulos, 1992). It is the first step for system developers to understand and describe the conceived or the real world system in information system (IS) analysis and design. A conceptual-modeling grammar is the language used to generate conceptual models. It provides a set of constructs and rules that show how to combine the constructs to model real-world domains (Wand & Weber, 2002). A conceptual modeling grammar should be based on a theory of ontology—a theory that articulates those constructs needed to describe the structure and behavior of the world in general (Wand & Weber, 1993;
Upper-level Ontologies help clarify the semantics of a conceptual modeling grammar and enhance its expressive power to capture problem domain knowledge precisely and unambiguously.

The precision, unambiguity, coherence, and expressive power of conceptual grammars broadly address two fundamental requirements in conceptual grammar development: soundness and completeness. While precision, unambiguity, and coherence address the soundness issue, the expressive power of a conceptual grammar is a measure of completeness of the grammar. Soundness of a grammar can be ensured by its careful design, but universal completeness is generally not attainable. Conceptual modeling grammars may only be boundedly complete in the sense that their expressive strength is adequate to satisfy most requirements within a bounded universe of discourse (Kishore, Sharman, & Ramesh, 2004). The notion of ontological expressiveness and a formal approach to assess ontological expressiveness of conceptual modeling grammars have been proposed by Wand and Weber (1993, 2004), and have been used by several researchers in the past (e.g., Milton, 2004; Green & Rosemann, 2000, 2004; Wand, Storey, & Weber, 2000).

The goal of this article is to elaborate the semantics of a recently developed conceptual modeling grammar from an ontological expressiveness perspective. The grammar, termed MibML (Multiagent-based Integrative Business Modeling Language), provides fundamental constructs, relationships, and axioms specially developed for systems analysis and design in the MIBIS (multi-agent-based integrative business information system) universe (Zhang, Kishore, Sharman, & Ramesh, 2004, 2005). Nevertheless, there is a need to understand the clear ontological semantics of the MibML grammar in order to apply it for conceptual modeling in a problem domain. As stated above, conceptual precision, unambiguous definitions, coherence of conceptual structures, and expressive power of the semantics are central to capturing problem domain knowledge correctly into MibML conceptual models. Otherwise, conceptual modeling could become arbitrary and the mapping of domain knowledge into modeling constructs could become highly dependent upon the beliefs, knowledge, and prior experience of system analysts. For example, the MibML grammar includes both goal and task as foundation constructs. Without a precise, unambiguous, and coherent denotation of these constructs, it may be difficult to model an instance such as “order inventory” as a goal or as a task. This problem of semantic ambiguity (clarity) is common in many conceptual grammars including the ER modeling formalism, which has recently been addressed by Wand et al. (2000). We believe such difficulties can be overcome by providing ontological semantics of the MibML constructs and their relationships via ontological analysis of the grammar.

In this article, we follow recent work in this area and apply the upper-level ontology of the Bunge-Wand-Weber (BWW) model (Wand, 1996) to interpreting the ontological expressiveness (Wand, 1996; Wand & Weber, 1993) of the MibML grammar. The proposed semantic mapping between the MibML grammar and the BWW model will help: (1) overcome difficulties and confusion for system analysts in selecting correct grammatical constructs to represent real-world phenomena; (2) facilitate knowledge sharing and reuse, and thus make it easy to integrate conceptual models from different developers; and 3) improve com-
munication between system analysts and other stakeholders such as users.

The remainder of this article proceeds as follows. First, we discuss the MIBIS universe and the MibML meta-model. Next, we describe the Bunge-Wand-Weber (BWW) model. We then explain the ontological approach of analyzing IS modeling grammars and how it is applied in our analysis of MibML. Following this we discuss the ontological analysis of MibML in the context of the BWW model. Finally, we conclude the paper and discuss future directions.

THE MIBIS UNIVERSE AND THE MIBML META-MODEL

Multiagent technology is being widely utilized for developing enterprise integration applications, such as in the areas of business process management (Blake & Gomaa, 2005; Huhns & Singh, 1998; Jennings et al., 1996; Jennings, Norman, Faratin, O’Brien, & Odgers, 2000), supply chain management (Huhns & Stephens, 2001; Salam, Singh, & Iyer, 2005; Smith, & Sadeh, 1998; Swaminathan, Smith, & Sadeh, 1998), and enterprise modeling (Lin, Tan, & Shaw, 1999; Pan & Tenenbaum, 1991; Sikora & Shaw, 2002). This phenomenon is partly driven by the fact that coordination is at the heart of both business integration applications and multiagent systems. While one of the fundamental requirements in business integration is that of coordination, multiagent systems are essentially coordination models and implement a number of coordination mechanisms (Kishore, Zhang, & Ramesh, Forthcoming). Such integrative application systems developed using multiagent technologies have been termed multi-agent-based integrative business information system (MIBIS) applications and they constitute a special class of information systems and a bounded universe of discourse (Kishore, Zhang, & Ramesh, 2004; Kishore et al., forthcoming).

A MIBIS system regards software agents as fundamental units to support integration of multiple distributed and decentralized work systems. Such a system is more flexible than a traditional workflow system in that it does not require all control flows to be specified at the system design time. Agents not only react to stimuli and changes in the system and the environment, but also adapt their behaviors and interactions with others based on their knowledge and information received at the run time in order to achieve their individual goals. Furthermore, a MIBIS system overcomes a workflow system’s inability to control resources involved in business processes. For example, a task in a traditional workflow system may be delayed if the person supposed to perform the task is on vacation. In MIBIS, agents are defined by roles, and a task can be performed by any agent playing a similar role. In short, a MIBIS system is role-centric and goal-oriented, and exhibits autonomous behavior.

In order to fully understand the MIBIS universe, Goal, Role, Interaction, Task, Information, Knowledge, Resource and Agent need to be captured in MIBIS conceptual models (Kishore et al., Forthcoming). Unfortunately, the traditional general-purpose conceptual modeling grammars, such as ER, UML, and workflow modeling languages, lack enough expressive power to capture completely these unique characteristics of MIBIS systems at the systems analysis level. Entity-relationship (ER) modeling ignores behavioral perspectives of entities. Object orientation such as UML provides no explicit support for agent concepts such
as autonomy, reactivity, and sociability (Franklin & Graesser, 1996; Odell, 2002; Wooldridge, 2002) and it is, therefore, difficult to model agent knowledge and interactions within MIBIS systems using OO modeling formalisms. Most enterprise and workflow models lack the essential semantics to concisely represent specific activities, tasks, business processes, business goals, and organization structures of a business (Scheer, 1999; Weigand & Heuvel, 2002). To respond to such emerging needs, MibML is developed to refine and formally define the above constructs, their relationships and axioms (Zhang et al., 2004). While resource was identified as a foundational ontological construct for the MIBIS universe, the MibML grammar does not include this construct because in a system’s modeling context, information about resources, rather than resources themselves, is a matter of concern.

A MIBIS is an organization of agents working together to accomplish business goals. As depicted in Figure 1, agents play roles. A role is essentially an abstraction for the tasks that are necessary to be performed and/or the interactions that need to occur to achieve individual agent goals, the information that needs to be accessed or will be generated during the course of performance of those tasks/interactions, and the knowledge that is needed for the successful execution of tasks and interactions and achievement of the goals.

**THE BWW MODEL**

The BWW model is an ontology of information systems proposed by Wand and Weber (1989, 1990, 1993) based on Bunge’s philosophical ontology (Bunge, 1977, 1979). The BWW model can be used to analyze the meaning of grammatical modeling constructs and has been applied for evaluating and improving notations and semantics of various conceptual modeling grammars. Wand et al. (2000) apply the BWW model in analyzing ER modeling constructs. Their analysis not only provides precise definitions of ER modeling constructs, but also derives rules for the use of relationships in ER conceptual modeling. Weber and Zhang (1996) examine and indicate the ontological deficiencies of Nijssen information analysis method (NIAM) using the BWW model. Green and Rosemann (2000) use the BWW model to analyze the five views—process, data, func-

![Figure 1. The MibML meta-model](image-url)
tion, organization and output—provided in the architecture of integrated information systems (ARIS). Their analysis indicates potential problems in representing all required business rules, specifying the scope and boundaries of the system, and employing a “top-down” approach to analysis and design. In addition, the BWW model is applied to examining Data Flow Diagrams (Wand & Weber, 1989), object-oriented modeling (Evermann & Wand, 2001; Opdahl & Henderson-Sellers, 2004; Parsons & Wand, 1997; Takagaki & Wand, 1991), and reference models in information systems development (Fettke & Loos, 2003).

This article applies the BWW model to interpret the semantics of the MibML constructs. Table 1 explains the BWW

<table>
<thead>
<tr>
<th>Ontological constructs</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BWW-thing</td>
<td>The elementary unit in the BWW model. The real world is made up of things. A composite thing may be made of other things.</td>
</tr>
<tr>
<td>BWW-properties</td>
<td>Things possess properties. A property is modeled via a function that maps the thing into some value. A property of a composite thing that belongs to a component thing is called a hereditary property. Otherwise it is called an emergent property. A property that is an inherent property of an individual thing is called an intrinsic property. A property that is an inherent property of two or more things is called a mutual property. A complex property comprises other properties, which may themselves be complex.</td>
</tr>
<tr>
<td>BWW-state</td>
<td>The vector of values for all property functions of a thing.</td>
</tr>
<tr>
<td>BWW-state law</td>
<td>A property that restricts the values of the properties of a thing to a subset that is deemed lawful because of natural or human laws.</td>
</tr>
<tr>
<td>BWW-event</td>
<td>A change of state of a thing. It is affected via a transformation.</td>
</tr>
<tr>
<td>BWW-transformation</td>
<td>A mapping from one state to another state.</td>
</tr>
<tr>
<td>BWW-transformation law</td>
<td>A property that defines the allowed changes of state.</td>
</tr>
<tr>
<td>BWW-history of a thing</td>
<td>The chronologically ordered states that a thing traverses in time.</td>
</tr>
<tr>
<td>BWW-coupling</td>
<td>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.</td>
</tr>
<tr>
<td>BWW-system</td>
<td>A set of things is a system if, for any bi-partitioning of the set, couplings exist among things in the two subsets</td>
</tr>
<tr>
<td>BWW-system environment</td>
<td>Things that are not in the system but interact with things in the system</td>
</tr>
<tr>
<td>BWW-level structure</td>
<td>Defines a partial order over the subsystems in a decomposition to show which subsystems are components of other subsystems or the system itself.</td>
</tr>
<tr>
<td>BWW-stable state</td>
<td>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.</td>
</tr>
<tr>
<td>BWW-unstable state</td>
<td>A state that will be changed into another state by virtue of the action of transformations in the system.</td>
</tr>
<tr>
<td>BWW-external event</td>
<td>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.</td>
</tr>
<tr>
<td>BWW-internal event</td>
<td>An event that arises in a thing, subsystem, or system by virtue of lawful transformations in the system</td>
</tr>
<tr>
<td>BWW-class</td>
<td>A set of things that can be defined via their possessing a particular set of properties</td>
</tr>
</tbody>
</table>

Table 1. Basic constructs in the BWW model (Adopted from Bunge, 1977 and Wand, Monarchi, Parsons, & Woo, 1995)

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
Figure 2. Ontological deficiencies of a conceptual model grammar

- **Ontological incompleteness** or construct deficit occurs when ontological constructs do not have equivalent constructs in the conceptual modeling grammar (Figure 2(a)). A modeling grammar is complete if all grammar constructs perfectly map to the ontological constructs.

- **Construct redundancy** is where several constructs from the conceptual-modeling grammar map onto a single ontological construct (Figure 2(b)).

- **Construct overload** is where several ontological constructs are mapped onto a single construct in the grammar (Figure 2(c)).

- **Construct excess** is where a grammatical construct might not map to any ontological construct (Figure 2(d)).

The semantics of the MibML grammar can be elaborated by assessing its ontological expressiveness. As depicted in Figure 3, we regard a MIBIS system as a model of the real world integrative business system. Accordingly, the MibML grammar should include necessary grammatical constructs to represent the ontological constructs that are used to describe the integrative business systems in the real world domain. In this article, we seek to identify ontological principles that will ensure that construct
excess, construct overload and construct redundancy are avoided in the grammar. This will not only reduce the semantic ambiguity of MibML, it will also help amplify the semantics of the MibML grammar. We do not focus upon ontological incompleteness because while upper-level ontologies in the philosophy discipline are generally developed to embody concepts for the entire universe, MibML is a limited modeling grammar designed to model only certain necessary aspects of the MIBIS bounded universe of discourse. For this reason we do not expect all ontological constructs to be completely mapped onto MibML constructs and believe that the MibML grammar will exhibit ontological incompleteness or construct deficit. This is in consonance with recent findings in the literature which indicate that some of the constructs of the BWW model, such as conceivable state space, conceivable event space, and lawful event space, do not have equivalent representations in the modeling grammars examined so far (Green & Rosemann, 2000).

**ONTOGOICAL ANALYSIS OF THE MIBML GRAMMAR**

In this section, we map the MibML constructs into the BWW model and thereby the meaning of the MibML constructs is defined precisely in terms of what it presents in the problem domain. Table 2 summarizes the interpretation of the MibML constructs. In this section, we prefix “MibML” and “BWW” to the grammatical constructs and the ontological constructs, respectively, to distinguish them clearly. We also follow the notations in (Wand et al., 2000) to represent ontological semantics of the MibML constructs in either propositional forms or functional forms. Please note that propositional forms and functional forms can be mutually converted. For example, assume $P$ is a set of persons and $C$ is a set of companies. Employment $E$ can then be represented as a propositional form:

$$E: P \times C \rightarrow \text{person } p \text{ works for company } c,$$

where $p \in P$ and $c \in C$ or as a function form $E: P \rightarrow C$. 

---

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
<table>
<thead>
<tr>
<th>The MibML Construct</th>
<th>BWW Interpretation and Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MibML-agent</td>
<td>A BWW-thing within the scope of the application</td>
</tr>
<tr>
<td>MibML-external entity</td>
<td>A BWW-thing outside the scope of the application</td>
</tr>
<tr>
<td>MibML-role</td>
<td>A complex BWW-binding mutual property between a BWW-thing representing a MibML-agent and a BWW-system representing a MIBIS system</td>
</tr>
<tr>
<td>MibML-goal</td>
<td>A stable state of a BWW-thing representing a MibML-agent or a MIBIS system</td>
</tr>
<tr>
<td>MibML-goal tree</td>
<td>A level structure of BWW-stable states that represent MibML-goals</td>
</tr>
<tr>
<td>MibML-individual goal</td>
<td>A stable state of a BWW-thing representing a MibML-agent.</td>
</tr>
<tr>
<td>MibML-goal assignment to role</td>
<td>A kind of BWW-state law that constrains the assignment of MibML-goals to MibML-roles</td>
</tr>
<tr>
<td>MibML-knowledge</td>
<td>A kind of BWW-complex property that consists of MibML-facts, MibML-deduction rules, MibML-activity execution structure, and MibML-activity execution constraint</td>
</tr>
<tr>
<td>MibML-fact</td>
<td>A kind of intrinsic BWW-property that represents a MibML-agent’s beliefs on other MibML-agents and the MIBIS environment</td>
</tr>
<tr>
<td>MibML-deduction rule</td>
<td>A kind of BWW-state law that governs the possible values of new MibML-facts (BWW-emergent properties) derived from existing MibML-facts (existing BWW-properties)</td>
</tr>
<tr>
<td>MibML-execution structure</td>
<td>A collection of BWW-transformation laws that constrain the order of performing MibML-tasks inside a MibML-agent</td>
</tr>
<tr>
<td>MibML-execution constraint</td>
<td>A collection of BWW-transformation laws that constrain how MibML-tasks between different MibML-agents should be performed</td>
</tr>
<tr>
<td>MibML-event</td>
<td>A BWW-event</td>
</tr>
<tr>
<td>MibML-external event</td>
<td>A kind of BWW-external event that emanates from the environment of a MibML-agent</td>
</tr>
<tr>
<td>MibML-temporal event</td>
<td>A kind of BWW-external event based on time passage consideration</td>
</tr>
<tr>
<td>MibML-state event</td>
<td>A BWW-internal event</td>
</tr>
<tr>
<td>MibML-interaction</td>
<td>A group of coupled BWW-events such that when a BWW-event in one thing changes a BWW-binding mutual property, it induces a BWW-event in another thing.</td>
</tr>
<tr>
<td>MibML-speech act</td>
<td>A BWW-event which is able to change a binding mutual property of a BWW-thing.</td>
</tr>
<tr>
<td>MibML-message</td>
<td>The BWW-binding mutual property that is changed when a MibML-interaction takes place.</td>
</tr>
<tr>
<td>MibML-task</td>
<td>A BWW-transformation of a BWW-thing</td>
</tr>
<tr>
<td>MibML-task input</td>
<td>A kind of BWW-transformation law to govern a MibML-agent’s state before a MibML-task can be performed</td>
</tr>
<tr>
<td>MibML-task output</td>
<td>A kind of BWW-transformation law to govern a MibML-agent’s state after a MibML-task is performed</td>
</tr>
<tr>
<td>MibML-task method</td>
<td>A kind of BWW-transformation law to specify how a MibML-task should be performed</td>
</tr>
<tr>
<td>MibML-partOf relationship between tasks</td>
<td>A BWW-transformation is a component of another BWW-transformation.</td>
</tr>
<tr>
<td>MibML-isa_typeof relationship between tasks</td>
<td>A BWW-transformation is a specialization of another BWW-transformation.</td>
</tr>
<tr>
<td>MibML-information entity</td>
<td>A BWW-thing that represents a passive business object</td>
</tr>
<tr>
<td>MibML-information flow</td>
<td>A BWW-transformation law that governs pre- or post-conditions of a MibML-task execution</td>
</tr>
</tbody>
</table>
Agent, Role, and External Entity

A MIBIS system is composed of interacting MibML-agents that work together in order to achieve a common system goal. MibML-agents are independent problem-solving units. They are software programs that are capable of perceiving their environment and taking autonomous actions in order to meet their design objectives. In the real world, a business organization consists of things such as human actors, computer systems, and machines to process business transactions. Therefore, we propose that MibML-agents represent a kind of BWW-thing within the scope of the MIBIS application in the context of the BWW model.

Just like a human actor plays a role in a business organization, a MibML-agent plays a MibML-role in a MIBIS system. A MibML-role defines the responsibilities and the privileges of the MibML-agent. It determines what MibML-tasks and MibML-interactions should be performed and what MibML-information and MibML-knowledge are accessible for the MibML-agent. In the context of the BWW model, we therefore propose that a MibML-role represents a complex BWW-binding property between a MibML-agent and the MIBIS system. Since a MibML-role is used as a template to define MibML-agents in the grammar, the definition of MibML-roles can be represented as a functional schema:

\[ R : I \times T \times P \times K \rightarrow \text{Functional schemata of MibML-roles} \]

where \( I \) is a set of MibML-interactions \( I = \{i_1, \ldots, i_s\} \); \( T \) is a set of MibML-tasks \( T = \{t_1, \ldots, t_k\} \); \( P \) is a set of privileges to access necessary MibML-information \( P = \{p_1, \ldots, p_n\} \); and \( K \) is a set of MibML-knowledge \( K = \{k_1, \ldots, k_m\} \).

Goal

MIBIS is a goal-oriented system. In the early stages of MIBIS development, a MibML-system goal is identified, decomposed iteratively, and structured into a MibML-goal tree to capture user requirements at different levels of details. The MibML-individual goals at the leaf level are then assigned to MibML-roles for MibML-agents playing these roles to accomplish.

In the context of the BWW model, a MibML-goal can be understood as a stable state which a MibML-agent or a MIBIS system tries to reach. For example, a goal “to create new customer order” indicates a state where the new customer information has been recorded and the new order has been created. Before a MibML-agent reaches its goal it will keep changing its state by performing necessary MibML-interactions and MibML-tasks. Correspondingly, a MibML-goal tree represents a level structure of BWW-stable states that represent MibML-goals.

Each MibML-role is assigned a MibML-individual goal. MibML-agents which play a certain MibML-role are responsible for accomplishing the MibML-goal assigned to the MibML-role. Such MibML-goal assignment constrains values of MibML-goals and MibML-roles for particular MibML-agents. For example, if
a MibML-agent plays MibML-role “salesperson”, then it must accomplish MibML-goal “to create new customer order.” In this sense, MibML-goal assignment represents a BWW-state law of a BWW-thing that represents a MibML-agent. Correspondingly, MibML-agents can be represented as a functional schema:

\[ A : R \times G \rightarrow \text{Functional schemata of MibML-agents} \]

where \( R \) is a set of MibML-roles \( R = \{ r_1, \ldots, r_i \} \); and \( G \) is a set of MibML-goal \( G = \{ g_1, \ldots, g_j \} \).

**Interaction**

A MibML-interaction, as a coordination mechanism to resolve interdependencies among MibML-agents, is defined as a sequence of MibML-speech acts. MibML-speech acts are utterances that contain information needed to assert and perform actions and serve as building blocks of communication protocols. They are used to perform actions such as booking, complaining, forgiving, etc. Associated with MibML-speech acts, MibML-messages are used to pass information between MibML-agents.

In the context of BWW model, a direct coupling between two or more BWW-things is caused by one or more BWW-binding mutual properties that all the BWW-things possess. Whenever one BWW-thing changes a BWW-binding mutual property, a corresponding BWW-event is induced in each of the other BWW-things. In this sense, MibML-messages represent BWW-binding mutual properties to be changed. MibML-speech acts represent BWW-external events that change BWW-binding mutual properties of message-receiving MibML-agents.

Correspondingly, a MibML-interaction represents a sequence of BWW-external events that change BWW-binding mutual properties of MibML-agents.

For example, in a supply chain system, a buyer agent can induce a BWW-external event (MibML-speech act “request”) in a supplier agent by changing BWW-binding mutual property “price quote request” which is possessed by both MibML-agents. In response, the supplier agent can induce a BWW external event (MibML-speech act “assertion”) in the buyer agent by changing BWW-binding mutual property “price quote.” In MibML, both events of requesting price quote and returning price quote make up a MibML-interaction.

Interaction patterns in MibML are described in MibML-roles. Therefore, the above interpretation can be defined as follows.

Let \( \text{Msg} \) represent MibML-messages passed between MibML-agents playing different MibML-roles. \( \text{Msg} \) is a functional mapping from one MibML-role to another:

\[ \text{Msg} : R_i \rightarrow R_j \]

where \( R_i (i = 1, 2) \) represents a set of roles.

Let \( \text{SA}_i \) represent MibML-speech acts induced in MibML-agents playing role \( R_i \). \( \text{SA}_i \) is a functional mapping of MibML-agents’ states and MibML-messages passed between MibML-agents:

\[ \text{SA}_i : S_{i1} \times S_{i2} \times \text{Msg} \rightarrow \text{Statements of SA}_i \]

where \( S_{i1} \) and \( S_{i2} \) are states of MibML-agents playing role \( R_i (i = 1, \ldots n) \).

The pattern of a MibML-interaction then is a functional mapping from a
Cartesian product of MibML-roles into a Cartesian product of their MibML-speech acts:

\[ I : R_i \times R_j \rightarrow SA_j \times SA_i \]

**Task**

A MibML-task is an activity performed solely by a MibML-agent. It is defined by MibML-input, MibML-output, and MibML-method. The MibML-inputs include facts from MibML-agents’ declarative knowledge component and data flows from databases and external entities; the MibML-outputs are task generated, and could result in updates or may even cause data flows within the MIBIS application or to external users; the MibML-method embodies the procedural knowledge used, which specifies the detailed logic for execution of the MibML-task.

MibML-inputs and MibML-outputs describe the pre-conditions (the state before task execution) and the post-conditions (the state after task execution) of a MibML-agent. That is, execution of a MibML-task transforms a MibML-agent from an input state to an output state. In the context of the BWW model, a MibML-task represents an internal BWW-transformation within a MibML-agent. MibML-inputs, MibML-outputs, and MibML-method represent BWW-transformation laws because they put constraints on task execution. For example, before a MibML-agent playing a supplier role can perform “generate price quote” task, it must be in a state where “inventory” information is known. After the task is performed, the agent is transformed to a new state in which price quote is generated.

Let \( T \) be a set of MibML-tasks of a MibML-agent and \( S \) be a set of possible states of the agent. Then:

\[ T : S \times S \rightarrow s_k \] is transformed to \( s_k \) following a predefined method

where \( s_j, s_k \in S \), and \( s_j \) and \( s_k \) represent the initial and the final state of task execution.

A MibML-task can be decomposed recursively into a task hierarchy along dimensions of subpart and subtype. A subpart of a MibML-task \( t \) is a constituent task of \( t \), which reflects the composition relationship between MibML-tasks. A subtype of a MibML-task \( s \) is a specialization of the task \( s \). The MibML-task hierarchy describes task interdependencies within a MIBIS application. For example, if MibML-task \( a_1, a_2, \) and \( a_3 \) are subparts of MibML-task \( a \), then execution of MibML-task \( a \) requires MibML-task \( a_1, a_2, \) and \( a_3 \) all are executed. In this light, both task subpart and task subtype in MibML represent a kind of BWW-transformation laws.

**Information**

MibML-Information refers to data resources available within a MIBIS application. It consists of MibML-information entities and MibML-information flows.

MibML-information entities refer to internal data within the system that are part of data stores. They represent regular business objects (such as PRODUCT, CUSTOMER, etc.) and other materialized views of data. The schema of MibML-information entities includes the structure of tables, the relationships, entity integrity constraints, referential integrity constraints, cardinality constraints, etc. MibML-Information entities may be implemented using relational databases and the schema corresponds to items typically stored in database repositories. The above explanation indicates that MibML-information entity is a complex construct which covers most concepts in
ER modeling such as entity, relationship, cardinality, etc.

In the context of the BWW model, there is not a single ontological construct that can be mapped to MibML-information entity. On the one hand, MibML-information entity represents BWW-things (e.g., PRODUCT, CUSTOMER); on the other hand, it is also used to represent BWW-mutual properties (e.g., the relationship between PRODUCT and CUSTOMER) and laws of BWW-things (e.g., cardinality constraints of PRODUCT and CUSTOMER). Therefore, MibML-information entity suffers construct overload. In addition, MibML-information entity also suffers construct redundancy when used as a representation of BWW-things. It has an overlap with the concept of MibML-external entity (e.g., both can be used to represent CUSTOMER). Construct overload and construct redundancy of MibML-information entity may result in confusion of developers whether a business domain phenomenon should be modeled as an external entity, as an information entity, or as a property.

It is a design consideration to include database-related aspects such as the table structure and cardinality as part of the MibML-information entity specification. Conceptual models should include only business-level entities to reflect the real-world business systems. In order to reduce semantic ambiguity, we want to confine the semantics of MibML-information entity to representing only business objects in business organizations. Such business objects can be physical business entities (e.g., inventory, machine, etc.) or conceived business entities (e.g., bank account, order, etc.). Therefore, we propose that MibML-information entities are mapped to BWW-things that represent passive business objects in the context of the BWW model. These business objects are passive in the sense that they do not initiate MibML-information flows.

MibML-Information flows represent data that are in transit; for example, they represent data moving between external users and MibML-agents, between MibML-information entities and MibML-agents, or between other external systems and MibML-agents. When MibML-information flows between a MibML-agent and an entity exist, it indicates that the MibML-agent requires data inputs to perform its tasks or it generates data outputs. Data inputs govern whether a MibML-task can be executed or not (e.g., task “create a new order” can only be executed when the specific customer data have already been transmitted to the MibML-agent.) On the other hand, data outputs govern which task or tasks should be executed (e.g., if data output “a new order” is required, only tasks necessary to create a new order should be executed.) In this sense, in the context of the BWW model, MibML-information flows represent BWW-transformation laws.

In addition, MibML-agents must possess necessary privileges (e.g., read, write, modify, delete, print, etc.) in order to send, receive, or process data. Such privileges are included in the MibML-role definition and represent mutual properties between a BWW-thing representing a MibML-agent and a BWW-thing representing a business object. Therefore, we have:

\[ P: R \times INF_e \rightarrow \text{role } r \text{ has privilege } p \text{ to process information entity } inf_e \]

where \( P \) is a set of information privileges, \( R \) is a set of roles, and \( INF_e \) is a set of information entities.
Knowledge

MibML-knowledge represents computational knowledge that describes MibML-agents’ beliefs about their environment and the constraints that govern their behaviors. It consists of MibML-facts, MibML-deduction rules, MibML-activity execution structure, and MibML-activity execution constraints, and is conceptualized to exist within individual MibML-agents. In this sense, MibML-knowledge represents a complex property of the BWW-thing that represents a MibML-agent in the context of the BWW model. Therefore, MibML-knowledge is defined as a functional mapping from a MibML-role into a Cartesian product of value sets of MibML-facts, MibML-deduction rules, MibML-activity execution structure, and MibML-activity execution constraints:

\[ K : R \rightarrow V_F \times V_L \times V_T \times V_X \]

where \( K \) is a set of MibML-knowledge, \( R \) is a set of MibML-roles, \( V_F \) is a value set of MibML-facts, \( V_L \) is a value set of MibML-deduction rules, \( V_T \) is a value set of MibML-activity execution structure, and \( V_X \) is a value set of MibML-activity execution constraints.

Both MibML-facts and MibML-deduction rules are declarative knowledge (know-that). MibML-facts are beliefs that a MibML-agent keeps about itself, about other MibML-agents in the system, and about the environment it resides in. Clearly, a MibML-fact is an intrinsic property of a MibML-agent. In general, MibML-facts are defined as:

\[ F : R \rightarrow V_F \]

where \( R \) is a set of MibML-roles, and \( V_F \) is a value set of MibML-facts.

MibML-deduction rules empower MibML-agents to engage in deductive reasoning. They allow MibML-agents to use existing information and facts to form new facts. Since new facts are derived from existing facts and existing facts represent intrinsic properties of an agent, new facts represent emergent properties – properties whose values depend upon the values of other properties. Because MibML-deduction rules put constraints on values of new facts, they represent BWW-state laws in the context of the BWW model. In general, deduction rules can be represented as:

\[ L : R \times F_{new} \times INF_j \times F_{old_1} \times \ldots \times F_{old_n} \rightarrow \text{Statement of the restrictions on deriving new facts from existing facts and information entities} \]

where \( R \) is a set of roles, \( F_{new} \) is a set of new facts, \( INF_j \) is a set of information flows, and \( F_{old_i} \) (\( i = 1, \ldots, n \)) is a set of existing facts.

MibML-activity execution structure and MibML-activity execution constraints represent procedural knowledge. MibML-Activity execution structure relates to knowing the order in which activities need to occur. It also includes information regarding how frequently activities have to be performed and whether certain activities are optional. In other words, it constrains how activities should proceed. In the context of the BWW model, we propose that MibML-activity execution structure represents BWW-transformation laws.

MibML-activity execution constraints determine the MibML-events that trigger activities. The MibML-event can be a MibML-external event, a MibML-temporal event, or a MibML-state event. MibML-external events emanate either from the environment including other MibML-agents...
or from end-users (such as a customer placing an order). MibML-temporal events are constraints placed on the execution of MibML-tasks or MibML-interactions based on some time consideration such as the lapse of some time period. A MibML-state event is an event that occurs inside a MibML-agent, and changes the state of the MibML-agent, and accordingly triggers a MibML-task or a set of MibML-tasks. The mapping between MibML events and BWW constructs are straightforward. A MibML-event corresponds to a BWW-event. Both MibML-external event and MibML-temporal event represent the BWW-external event. MibML-state event represents the BWW-internal event.

Let $EV$ be a set of MibML-events, $S$ be a set of possible states of a MibML-agent and $C$ be a set of causes that result in the MibML-agent state transitions. Then:

$$EV: S \times S \times C \rightarrow c \text{ causes a state transition of the MibML-agent from } s_j \text{ to } s_k$$

where $s_j, s_k \in S$, and $c \in C$. $C$ may be a set of actions of MibML-external entities and other MibML-agents in the system, a set of values of time passage, or a set of MibML-tasks performed by the MibML-agent itself. Correspondingly, these different types of causes result in MibML-external events, MibML-temporal events, and MibML-state events.

**Discussion**

This study is a further development of the MibML grammar to clarify its ontological semantics in the context of the BWW ontology. The clarification is summarized in Figure 4. The MibML grammar includes three types of things: agents, information entities, and external entities. Agents interact with each other via interactions defined in speech acts (represented as double-headed solid lines between agents to indicate that they change binding mutual properties of agents.) Agents also communicate with external entities and information entities via information flows for task execution (represented as double-headed dotted lines connecting to tasks to indicate that information flows either provide inputs to tasks or are generated as outputs of tasks and thus

*Figure 4. Ontological semantics of MibML*
Agents require information privileges to communicate with information entities (represented as double-headed dashed lines between agents and information entities to indicate they are mutual properties.) The functional scheme of agents is determined by goal and role. A goal is a stable state an agent tries to reach. A role defines an agent’s intrinsic properties (knowledge including facts, deduction rules, activity execution structure, and activity execution constraints), mutual properties (interactions and information privileges), and internal transformation (tasks).

The definitions of the ontological semantics of the MibML grammar given above are quite clear and they will enhance user communication, and will benefit MIBIS system developers greatly as they will be able to develop precise conceptual models for a MIBIS system. The clear semantics also make it easy for systems developers to identify correct MibML constructs to represent problem domain knowledge. Take goal and task for example. It is always difficult to differentiate a goal and a task. A real-world instance “order inventory” can either be modeled as a goal or as a task depending on the level of abstraction. With a goal clearly defined as a stable state of an agent and a task as an internal transformation, we understand that a task changes states of an agent. That is, if an instance generates data flows or modifies values of internal properties of an agent, it is modeled as a task. Therefore, if “order inventory” results in update or creation of inventory information, it is modeled as a task. Otherwise, it is modeled as a goal. Similarly, we can easily determine that a “customer” is modeled as an external entity, but a “product” is modeled as an information entity based on their ontological semantics.

Further, unambiguous ontological semantics will also help systems developers focus on business modeling without considering design and implementation details at the systems analysis stage. While the MibML grammar incorporates some design and implementation details in the construct specifications to facilitate MIBIS development during the design and implementation stages, these technical considerations result in complex conceptual models and make it increasingly difficult for users to understand the conceptual models. For example, the information entity specification in MibML includes table structure, referential constraints, etc. Therefore, these technical issues are not part of ontological semantics defined and it is not necessary for these details to be included in MIBIS conceptual models.

In summary, ontological analysis of the MibML grammar in this paper provides a starting point for developing ontological principles and step-by-step guidelines or a methodology for constructing MIBIS conceptual models.

CONCLUSIONS AND FUTURE WORK

MibML is a recently proposed conceptual-modeling grammar to respond to the emerging needs for a special-purpose conceptual-modeling grammar for developing multiagent-based integrative business information systems. To further develop the grammar, there is a need to amplify and clarify ontological semantics of the MibML constructs through ontological analysis. Such clarification not only provides a precise ontological definition for the MibML constructs, but also facilitates...
the mapping from business domain knowledge onto MibML constructs. In the paper, we have carefully examined the MibML constructs and clarified and amplified their ontological semantics in the context of the Bunge-Wand-Weber ontology.

Some future directions for development of the grammar includes the following areas: (1) further detailed ontological investigations of the MibML grammar which would lead to the identification and discovery of ontological principles for using MibML in conceptual modeling; (2) development of a systematic design methodology with step-by-step guidelines for MIBIS modeling (e.g. how to identify the scope of the model, how to identify relevant MIBIS entities, etc.) using MibML as the representation and analysis formalism; (3) experimental validation of the MibML rules through practical design evaluations; and (4) development of a CASE tool to enable automated MIBIS analysis and design and enforcement of MibML model consistency based on the ontological rules developed in this work. All these areas have strong research potential and significant practical value and we are currently investigating some of these areas.

ACKNOWLEDGMENTS

The authors are grateful to Kaushal Chari, participants at the MSS Colloquim at SUNY Buffalo, anonymous reviewers at the AMCIS 2004 conference, and the guest editors and anonymous reviewers at the Journal of Database Management for providing their invaluable comments and suggestions during the course of development of this paper. The article has improved much as a result of their comments.

REFERENCES


nal of Database Management, 15(2), 105-117.
Moulin & C. Rolland (Eds.), *Object oriented approach in information systems* (pp. 275-296). North, Holland: Elsevier Science.


Hong Zhang is an assistant professor of computer information systems at Missouri State University. His research interests are computational ontology, enterprise integration, system analysis and design, and multi-agent systems. His papers have been published or accepted for publication in Communications of the ACM, Decision Support Systems, and INFORMS Journal on Computing. He is an executive committee member of AIS SIG on Ontology-Driven Information Systems (SIG-ODIS).

Rajiv Kishore is an associate professor in the School of Management at the State University of New York at Buffalo. His primary research interest is in how organizations can improve their IT services delivery capabilities, particularly through IT outsourcing and globally distributed work, use of innovative systems analysis and design methods and techniques, and adoption of advanced technological innovations. His papers have been published or accepted for publication in Journal of Management Information Systems, Communications of the ACM, Decision Support Systems, Information Systems Frontiers, and Journal of Healthcare Information Management. Rajiv has presented his research at ICIS, HICSS, AMCIS, SIM, and so forth. He received a best paper award at AMCIS 2001 and was nominated for a best paper award at AMCIS 2003 and HICSS 2004. He is also the recipient of a multi-year National Science Foundation research grant as a co-principal investigator in the area of IT outsourcing. Rajiv has consulted with a number of large companies, some of which include BellSouth, Blue Cross Blue Shield of Minnesota, IBM, and Pioneer Standard Electronics.

Ram Ramesh is a professor and chair of management science and systems in the School of Management at the State University of New York at Buffalo. His research streams include conceptual modeling (ontologies, connectionist modeling and nonmonotonic reasoning, economics and technologies of internet capacity provision networks (CPN) and database systems and distributed computing frameworks. He is a founding board member of AIS SIG on Ontology-Driven Information Systems (SIG-ODIS) and a board member of AIS SIG on Semantic Web and Information Systems (SIG-SEMIS). He serves as an associate editor for INFORMS Journal on Computing, Communications of the AIS, and several more. He is co-editor-in-chief of Information Systems Frontiers, and he has edited volumes in Annals of OR and CACM. He is currently guest-editing an issue of the Journal of AIS on Ontologies in the context of information systems. He publishes extensively in journals such as INFORMS Journal on Computing, Information Systems Research, IEEE/TKDE, ACM/TODS, and CACM, among others.