Computational & Mathematical Organization Theory 6, 381–394, 2000.

Ontologies to Support Process Integration in Enterprise Engineering

MICHAEL GRÜNINGER, KATY ATEFI AND MARK S. FOX
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Canada M5S 3G8
email: gruninger@eil.utoronto.ca
email: atefi@eil.utoronto.ca
email: msf@eil.utoronto.ca

Abstract

Enterprise design knowledge is currently descriptive, ad hoc, or pre-scientific. One reason for this state of affairs in enterprise design is that existing approaches lack an adequate specification of the terminology of the enterprise models, which leads to inconsistent interpretation and uses of knowledge. We use the formal enterprise models being developed as part of the Toronto Virtual Enterprise (TOVE) project to provide a precise specification of enterprise structure, and use this structure to characterize process integration within the enterprise. We then use the constraints within the enterprise model to define a special class of enterprises, and discuss the concepts necessary to characterize process integration within this class. The results of this paper arose out of the successful application of these ontologies to the analysis of the IBM Opportunity Management Process in a joint project with IBM Canada.

Keywords: enterprise design, ontologies, information

1. Introduction

Enterprise design knowledge is currently descriptive, ad hoc, or pre-scientific. It is a collection of heuristics which are not applicable in all circumstances. There is no coherent theory of enterprise design which can explain why different approaches and techniques work for certain enterprises and why they fail for other enterprises. What is needed is a distillation of the principles for enterprise design implicit within the heuristics, and the formalisation of these principles as logical theories. This would allow a more scientific methodology in which we test, compare, and validate different enterprise design theories.

These concerns have led to the development of enterprise models (Fox and Gruninger, 1998). An enterprise model is a computational representation of the structure, activities, processes, information, people, behaviour, goals and constraints of a business, government or other enterprise. It can be both descriptive and definitional—spanning what is and what should be. The role of an enterprise model is to achieve model-driven enterprise design, analysis, and evaluation.

However, existing approaches to enterprise modelling lack an adequate specification of the semantics of the terminology of the underlying enterprise models, which leads to inconsistent interpretations and uses of knowledge. Analysis is hindered because models tend to be unique to the enterprise, and are rarely reused. Within enterprise operations, obstacles to interoperability arise from the fact that the legacy systems that support the functions in many enterprises, were created independently, and do not share the same semantics for the terminology of their enterprise models.

A rigorous foundation for enterprise design, analysis, and operations therefore requires a formal specification of the semantics of enterprise models through the use of ontologies. An ontology is a formal description of entities and their properties, relationships, constraints, behaviours. It provides a common terminology that captures key distinctions and is generic across many domains. We will use the ontologies being developed as part of the Toronto Virtual Enterprise (TOVE) project to provide a precise specification of enterprise structure, and use this structure to characterize process integration within the enterprise.

We will then use the constraints within the enterprise model to define a special class of enterprises, and discuss the concepts necessary to characterize process integration within this class.

2. TOVE Ontologies

The goal of the TOVE project has been the creation of an integrated set of ontologies to support enterprise modelling. This consists of a set of generic core ontologies, including an activity ontology that spans activity, state, time and causality (Fox and Gruninger, 1994), resource ontology (Fadel et al., 1994), an organization ontology spanning structure, roles, and communication (Fox et al., 1995), and a product ontology which includes features, parameters, assemblies, versions, and revisions (Baid, 1994), (Lin et al., 1996). It also includes a set of extensions to these generic ontologies to cover concepts such as cost and quality.[1]

The primary component of the ontology is its terminology for classes of processes and relations for processes and resources, along with definitions of these classes and relations. Such a lexicon of terminology along with some specification of the meaning of terms in the lexicon constitutes the ontology. The model theory of the ontology will provide a rigorous mathematical characterization of the terminology of ontology. The proof theory of ontology provides axioms for the interpretation of terms in the ontology.

The language of the ontology is a set of symbols (lexicon) and a specification of how these symbols can be combined to make well-formed formulae (grammar/syntax). The lexicon consists of logical symbols (such as connectives, variables, and quantifiers) and nonlogical symbols. The Knowledge Interchange Format (KIF) (Genesereth and Fikes, 1992) is the language which provides the logical lexicon for the TOVE ontologies. The nonlogical part of the lexicon consists of expressions (constants, functions, relations) that refer to concepts in some domain; for example, in a process ontology, this would include everything needed to describe processes.

This logical approach to ontology design and implementation serves two purposes. First, it allows a precise and rigorous characterization of the consistency and completeness of the ontology with respect to its intended application. Second, it supports the implementation of automated inference for any enterprise model that uses the ontology. This second property is crucial in the work presented in this paper. Using the set of logical constraints, we were able to automatically infer that there were problems with the business processes within the enterprise model.

We will briefly give an overview of the semantic foundations for the two ontologies which play a pivotal role within process integration, namely, the Activity and Organization ontologies.

2.1. Activity Ontology

The semantics of the TOVE Activity Ontology are based on the situation calculus (Reiter, 1991). Within the situation calculus, there are three sorts of objects: *actions, situations, and fluents*. Intuitively, a model of the axioms of the situation calculus represents a complete set of alternative activity occurrences. Each situation is the result of some linear sequence of action occurrences. Thus, the structure of situations is that of a tree; there exists an initial situation, and a function *do* (a, s) which is the name of the situation that results from performing action a in situation s. Thus, each branch that starts in the initial situation can be understood as a hypothetical future. The tree structure of the situation calculus shows all possible ways in which the events in the world can unfold. Any arbitrary sequence of actions identifies a branch in the tree of situations. For example, figure 1 shows a finite subset of a situation tree. Situation S_3 is the result of performing the sequence of actions *Validate, Select, Notify*, while situation S_6 is the result of performing the sequence of of actions *Validate, Notify, Select*.

The notion of a fluent captures intuitions about state; a fluent represents some property of the world whose value may change as the result of performing an action. Fluents are assigned to each situation in the tree; the intended interpretation of the relation *holds* (f, s) is that the fluent f is intuitively true in situation s. The preconditions which are required for an action to occur are defined with respect to the fluents which hold at a particular situation.

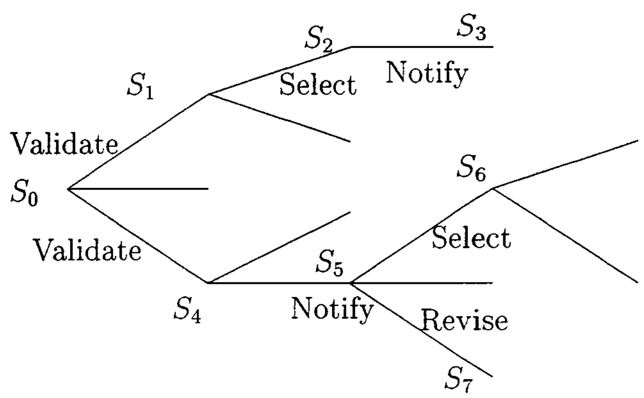

Figure 1. Situation trees.

The relation *poss* (a, s) specifies that an action a satisfies its preconditions in situation s and can therefore possibly occur.

We also need to be able to define complex aggregations of actions. This is done by identifying subtrees of situations within the situation tree; each subtree corresponds to different possible sequences of primitive actions which occur whenever the complex action occurs. For example, a *nondeterministic* action will consist of subtrees such that different primitive subactions occur on different branches. In figure 1, a nondeterministic action occurs in situation S_5—either the action *Revise* or the action *Select* occurs. A *partially ordered deterministic* action will consist of subtrees in which the same primitive subactions occur on each branch, but each branch corresponds to a different ordering of subaction occurrences. In figure 1, the two branches containing situation S_3 and S_6 specify a subtree for a partially ordered action—the same actions occur on these two branches, but *Select* and *Notify* satisfy different orderings. *A complex action definition* is a set of axioms which is satisfied by a set of subtrees within a situation tree.

2.2. Organization Ontology

The semantic foundations of the TOVE Organization Ontology (Fox et al., 1995) are an extension of the situation calculus. Whereas the situation tree specifies the set of all possible sequences of action occurrences, we will want to identify a subset of these sequences as those which an agent intends to occur; these sequences define an *intended subtree* of the situation tree for that agent. For example, in figure 1, we may specify that the two branches containing situations S_3 and S_6 are part of the intended subtree; in this case, the agent intends the actions on these branches to occur. *Intended constraints* are set of axioms which are satisfied in the intended subtree of an agent.

Agents are defined by the set of constraints on their behaviour. A *goal* is a future state which the agent intends to achieve; within the situation tree, this means that there exists a situation within the agent's intended subtree in which the fluent associated with the state holds. A *policy* is a constraint which requires that some fluent hold in all situations within the agent's intended subtree. *Obligations* and *responsibilities* are defined with respect to the actions which the agent intends to perform. Examples of this can be the commitment to perform an action (in which case there exists a situation in the agent's intended subtree which is the result of the occurrence of the action) or the obligation to perform an action under certain circumstances (in which case there will exist an occurrence of the action in the agent's intended subtree whenever the fluent associated with the condition holds). In figure 1, if the two branches containing situations S_3 and S_6 are part of the intended subtree for the agent, then the agent has the responsibility of performing the actions *Validate, Select* and *Notify*.

Group agents are defined by the assignment of the group's constraints and processes to the members of the group. For example, different members of the group may have different goals and consequently perform different subactivities of a process.

Finally, we need to be able to represent the knowledge and abilities of agents within an enterprise. Within the Organization Ontology, *knowledge* is treated as a fluent, so that we can model an agent's knowledge as its "internal" state. This allows the distinction between what an agent knows and what state actually holds in the world. Each process within the enterprise has knowledge preconditions, which specify what the agent needs to know in order to perform the process. The knowledge preconditions for the processes within the enterprise are achieved by knowledge-producing actions, that is, actions whose effects give the agent knowledge about the state of the world.

Ability is intuitively represented as "know-how"—an agent is able to perform an action if it knows the sequence of subactions which must occur as well as the preconditions required for each subaction.

3. What is an Enterprise?

An integrated enterprise model provides the language used to specify an explicit definition of an enterprise. Any enterprise model must have the expressiveness to capture the sets of constraints satisfied by the agents of the enterprises, the activities that they perform, and the resources required by these activities. In this section, we consider some of the ontologies which are needed to capture our intuitions about enterprise structure and process integration within an enterprise.

An explicit definition of an enterprise is essential for a precise characterization of enterprise design and analysis. For example, we need to explore alternative models in the design of enterprises spanning organization structure and behaviour. In order to reason about alternative designs for enterprises, we need to reason about different possible sets of constraints for enterprises within the language. We need to ask the questions—can a process be performed in a different way, or can we achieve some goal in a different way? Can we relax the constraints in the enterprise such that we can improve performance or achieve new goals?

We also need to be able to determine the impact of changes on all parts of the enterprise. For example, if we relax one of the policies, how will this affect the quality of products or services provided by the enterprise? If we purchase a new kind of machine, how will this affect the activities that are performed? Will we need to retrain people in the enterprise to give them the skills to use the machine? If we change the activities that are performed, how will this change resource consumption?

By representing the enterprise as a set of constraints using the ontologies, all of the above questions can be considered as either constraint satisfaction or logical entailment problems. All of the relationships among the different constraints within the enterprise are therefore made explicit.

The characterization of an enterprise as a set of logical constraints presupposes the existence of an adequate set of ontologies that define the concepts necessary for capturing the wide array of phenomena within an enterprise. The work presented in this paper builds on two existing enterprise modelling projects—the TOVE Ontologies (which have already been discussed) and the Enterprise Ontology, from the University of Edinburgh.

The Enterprise Project at the University of Edinburgh (Uschold et al., 1997) supports an environment for integrating methods and tools for capturing and analyzing key aspects of an enterprise, based on an ontology for enterprise modelling. This ontology (the Enterprise Ontology) has five top-level classes for integrating the various aspects of an enterprise:

- Meta-Ontology: Entity, Relationship, Role, Actor, State of Affairs
- Activities and Processes: Activity, Resource, Plan, Capability
- Organisation: Organisational Unit, Legal Entity, Manage, Ownership
- Strategy: Purpose, Strategy, Help Achieve, Assumption
- Marketing: Sale, Product, Vendor, Customer, Market

The Enterprise Ontology is semi-formal—it provides a glossary of terms expressed in a restricted and structured form of natural language supplemented with a few formal axioms using Ontolingua. For example, the following expression is the specification of the concept of "Plan" within the Enterprise Ontology:

```
(define-frame Plan
     :own-slots
     ((Documentation
     "The Activity-Spec in the Intended-Purpose Relationship")
      (Instance-Of Class)  (Subclass-Of Activity-Spec)
      :template-slots
       ((Intended-Purpose  (Minimum-Slot-Cardinality 0)
                           (Slot-Cardinality 1)
                           (Slot-Value-Type State-Of-Affairs)) )
      :axioms
       (<=>  (Plan ?plan)
             (exists (?soa)
                     (Intended-Purpose ?plan ?soa)))
       :issues
       ("This definition is equivalent to: An Activity-Spec that
         is associated with an Intended-Purpose."
         "This is a special Role-class.") )
```

In other words, a plan is an instance of the class of activity specifications, and is always associated with an intended purpose. The KIF axiom states that for every plan there exists a state-of-affairs that is the intended purpose of the plan.

Lloyd's Register has used the Enterprise Ontology for more effective modelling and re-engineering of business processes for strategic planning. IBM UK intends to exploit the Enterprise Ontology in modelling its own internal organisation as well as providing technical input via its Business Modelling Method BSDM (Business Systems Development Method).

However, because the Enterprise Ontology is not completely axiomatized within logic, it cannot be used to support automated reasoning. Since the application of automated inference to business process analysis is one of the primary motivations for our work, we will concentrate on the formal ontologies from TOVE. In the next section, we present a characterization of the constraints that model an enterprise, using the TOVE ontologies.

3.1. Structure of an Enterprise

Within the TOVE ontologies, an enterprise is defined by the following set of constraints represented as sentences in first-order logic:

$$\mathcal{E}_{action} \cup \mathcal{E}_{resource} \cup \mathcal{E}_{org} \cup \mathcal{E}_{goals} \cup \mathcal{E}_{products} \cup \mathcal{E}_{services} \cup \mathcal{E}_{occ} \cup \mathcal{E}_{external}$$

The set of sentences in $\mathcal{E}_{action}$ uses the TOVE Activity Ontology, and consists of the following set of constraints on activities within the enterprise:

- The complete set of activities that can possibly be performed within the enterprise. These activities correspond to the process plans of the enterprise, and the auxiliary activities that support them, such as setup and cleanup activities.
- The set of complex action definitions for the knowledge-producing activities in the enterprise.
- The set of knowledge precondition axioms for actions within the enterprise. These define what agents need to know in order to perform their activities.

The set of sentences in $\mathcal{E}_{resource}$ uses the TOVE Resource Ontology, and consists of the following set of constraints on resources required by activities within the enterprise:

- Constraints defining the complete set of resources that can possibly be used or consumed by activities in the enterprise.
- Properties of the resources related to the concurrency constraints by defining the capacities of the resources in the enterprise.
- Intended constraints on the state of resources in the enterprise. This includes constraints such as preventing spoilage and maintaining safety stock levels. For example, the following KIF sentence expresses the constraint that spoilage of a resource occurs at some fixed time depending on the shelf life of the resource and the time at which the resource was produced:

```
(forall (?a ?t ?d ?r)
(=>   (and   (occursT (terminate ?a) ?t)
             (produces ?a ?r)
             (shelf-life ?r ?d))
      (occursT (spoilage ?r) (+ ?t ?d))))
```

$\mathcal{E}_{org}$ is the set of sentences defining the constraints among organizational roles, positions, and agents within the enterprise. An example of this are occurrence constraints on the behaviour of agents. All of these sentences use the TOVE Organization Ontology.

$\mathcal{E}_{goals}$ defines the goals to be achieved by the enterprise. This is a set of existential intended constraints, that uses an extension of the TOVE Organization Ontology.

$\mathcal{E}_{products}$ is the set of sentences defining constraints on product fluents. This includes intended constraints for product design requirements, quality constraints, and product standards. All of these sentences use the TOVE Product Ontology.

$\mathcal{E}_{services}$ is the set of occurrence constraints on product and knowledge fluents that define the activities the enterprise performs as services.

$\mathcal{E}_{occ}$ is the set of occurrence constraints on activities in the enterprise. This includes policies and performance constraints, such as the following examples:

- When an order is made, a copy is sent to the regional office.
- All deliveries must be made within 15 minutes of placing the order. Within KIF, this would be expressed as:

```
(forall (?x ?t1 ?t2)
(=>    (and   (occursT (order-entry ?x) ?t1)
              (occursT (delivery ?x) ?t2))
       (< (- ?t1 ?t2) 15)))
```

All of these sentences use an extension of the TOVE Activity Ontology.

$\mathcal{E}_{external}$ is the set of constraints defining the external environment of the enterprise, dealing with customers, markets, suppliers, and competitors. It also includes the definitions of the activities performed by agents external to the enterprise (e.g. suppliers, subcontractors), but whose effects are required by the activities within the enterprise.

Using this framework, we can characterize classes of enterprises by sets of assumptions over their processes, goals, and organization constraints.

3.2. Enterprises and Process Integration

Process integration is defined by the relationship between the processes of the enterprise and the organizational structure and constraints of the enterprise. We can use the classes of constraints for enterprise structure defined in the preceding section to characterize process integration within an enterprise with respect to the goals of the enterprise and its customers.

We first need to specify the goals of the customers, since all enterprises must be committed to achieving the customers' goals. Additional goals for the enterprise are generated from these commitments. There are two classes of customer goals for an enterprise:

- The enterprise provides the customer with a product (a resource which did not exist before).
- The enterprise changes the fluents of existing resources or provides information. In this case, the enterprise provides a service.

If the goals of the enterprise are focussed on the delivery of products, this will require the constraints in $\mathcal{E}_{product}$. If the goals of the enterprise are based on the delivery of services, this will require the constraints in $\mathcal{E}_{services}$. Customers also have requirements on the products and services of the enterprise. The objectives of the enterprise must include satisfying these requirements. In addition, the enterprise must also satisfy standards that have been set for any products.

The second characteristic of process integration arises from imposing the following integrity constraint on the definition of the enterprise—all of the enterprise's goals must be achievable. This in turn defines constraints for the other aspects of the enterprise.

In particular, the enterprise must be able to perform those activities whose effects achieve the goals of the enterprise. These activities will have preconditions which must in turn be achieved. This leads to the definition of the activities of the enterprise and the constraints in $\mathcal{E}_{action}$. The preconditions of the enterprise's activities are defined in terms of the resources required to support the activities. This leads to the constraints in $\mathcal{E}_{resource}$.

The complex action definitions and constraints in $\mathcal{E}_{action}$ also require the definition of the necessary organizational roles to support the activities. In particular, this includes the authority and empowerment constraints for all of the agents in the enterprise. This leads to the constraints in $\mathcal{E}_{org}$.

The third characteristic of process integration considers the relationships between the enterprise and its environment. In addition to customers, there are other external organizational roles that generate a set of constraints that define the enterprise in the same way that customers do. These include the roles of supplier, competitor, and market. There are also internal organizational roles and positions that impose constraints on the definition of the enterprise; these include employee, management, and stockholder. The organization ontology must define these organizational roles and the constraints on their activities, providing the constraints in $\mathcal{E}_{org}$ and $\mathcal{E}_{external}$.

4. Business Process Ontology

The characterization of process integration from the previous section applies to generic enterprises, but we will also want to gain insight into other special classes of enterprises, as defined by their business processes and organizational constraints. This section reviews an application of the TOVE ontologies to the analysis of a process as part of a joint project with IBM Canada. This leads to the introduction of new concepts which extend the Activity and Organization ontologies and which comprise an initial ontology for business processes. For each characteristic of the class of enterprise which we are considering, we will discuss the necessary concepts that must be captured by the Business Process Ontology in order to support process integration.

4.1. Motivating Scenario

IBM has established a world-wide focus on a number of its key processes. One of these processes, the Customer Relationship Management process, is expected to play a significant role in redefining the relationship between IBM and its customers. The robustness of this process is essential for IBM to adapt to the constantly changing expectations of its customers while maintaining profitability. IBM Canada, a leader in integrated design and implementation of processes and tools that support process management, has been a key contributor to the design and implementation of the Customer Relationship Management process.

The Opportunity Management Process (OMP) is a key subprocess of IBM's Customer Relationship Management process which was analyzed using the TOVE ontologies in a joint project with IBM Canada (Atefi, 1997). OMP identifies an opportunity to sell a product or service to a customer, and determines whether or not IBM should pursue the opportunity. In particular, OMP has the following upper-level structure:

1. *Identify Opportunity*
   OMP is initiated when an IBM employee identifies an opportunity to sell a product or service to a customer. The employee's role within the process is the Opportunity Identifier, and it is the first contact point for the customer.
2. *Validate Customer Business Needs*
   In this activity, the Opportunity Identifier collects the customer's data and estimates the likelihood that IBM will be able to meet the customer's business requirements.
3. *Establish Opportunity Owner*
   If the Opportunity Identifier decides that IBM can fulfill the customer requirements, a Resource Coordinator within IBM assigns an employee as the Opportunity Owner. Once assigned and introduced to the customer, the Opportunity Owner will be the contact point for the customer.
4. *Qualify and Select/Deselect Opportunity*
   The Opportunity Owner and Opportunity Business Manager evaluate the opportunity, considering profitability and possible risks. If the opportunity is qualified, it will be selected and passed to another subprocess of the Customer Relationship Management process. If the opportunity is not qualified, the the customer will be notified.

4.2. Customer-Enterprise Interaction

The motivating scenario for OMP serves as the source of intuitions for the class of enterprises which must be modelled. We need a formal characterization of this class of enterprises before we can determine whether an enterprise has achieved process integration. Within the TOVE Business Process Ontology, we are considering enterprises which provide products and services for potential customers, and which have the following characteristics:

- Customer interaction processes
- Information flow
- Knowledge-producing evaluation processes
- Unintended behaviour

To support process integration for the class of enterprises in the motivating scenarios, we will need to define classes of business processes and the constraints which they impose on the enterprise's agents.

4.2.1. Customer Interaction Processes. The class of enterprises which we are considering is characterized by tightly coupled interactions between the customer and the enterprise. There must exist processes by which the enterprise acquires and commits to satisfying the customer's goals and requirements. This must be a dynamic relationship, since the enterprise must be able to respond to changes in the customer's requirements. Within this class of enterprises, there also exist processes which satisfy queries about the status of a customer request.

In the IBM Opportunity Management Process, the focus is on the various agents which are the contact between the enterprise and the customer; we can generalize this to the notion of an agent which is transaction contact, where a transaction is a specific customer opportunity.

Transaction Contact Existence. The key to process integration in this class of enterprises is the assignment of agents to the actions which they perform. Although the specification of a process may require that a contact agent exist, the dynamic nature of agent-action assignments within an enterprise can lead to problems in practice. Without an agent who is explicitly responsible for being the contact with the customer, the enterprise can lose the customer when assignments change. Thus, the problem of the existence of a transaction contact can be characterized by the following question:

Given the activities which assign and modify the assignments of transaction contacts, does there exist a situation in which no transaction contact is assigned to the transaction?

Within KIF, this question is equivalent to determining whether or not the following sentence is a logical consequence of the constraints within the enterprise model:

```
(exists (?act ?agent ?customer ?time)
(and  (subactivity ?act OMP)
      (occursT (terminate ?act) ?t)
      (holdsT (agent-constraint ?customer ?OMP) ?t)
      (not (holdsT (agent-constraint ?agent
                   (transaction-contact ?agent ?customer) ?t))))
```

If such a situation exists, then there is possibly a point during the occurrence of the process in which the transaction contact does not exist, which would violate one of our criteria. Thus, the Business Process Ontology defines the transaction contact in terms of its responsibilities and obligations. It also specifies the actions whose effects assign and modify assignments of agents and actions.

Transaction Contact Notification. There is another possibility which can lead to problems within the process

Does there exist a situation in which there exists a transaction contact, but the customer of the transaction does not know who it is?

For example, even if the enterprise is able to dynamically modify the agent-action assignments for the process, if the customer is not notified of the change, the enterprise will no longer be able to respond to changes in customer requirements, which was one of the criteria which the business process must satisfy if it is to be integrated with the organization structure.

Therefore, the Business Process Ontology must not only specify the actions whose effects assign and modify assignments of agents and actions, but it must also specify notification activities whose occurrence is triggered by the occurrence of assignments of agents and actions.

Transaction Traceability. Although the transaction contact is the agent responsible for answering queries about the status of a customer's request, it is not the same agent responsible for performing the activities that satisfy the customer's request. For example, if the customer requests a particular configuration of hardware and software solutions, there may be a team within the enterprise which determines the possible products and services which satisfy this request. Although the transaction contact may not be a member of this team, it must nevertheless know which team member is responsible for the various subactivities if it is to be able to determine the current status of the customer request.

Thus, given an enterprise model, we must be able to determine whether there exists any situation in which an agent is assigned to an activity in a transaction, but the transaction contact neither knows or is able to know to whom the activity is assigned.

To satisfy this criterion for the process integration, the Business Process Ontology must be able to represent the following:

- Define information resources where agent assignments are recorded and retrieved by the transaction contact.
- Define activities which report changes to agent assignments and record them in information resources.

4.2.2. Information Flow. The requirements for information flow can be characterized by the following question:

Who needs to know what, when do they need to know it, and how do they find out?

Since each process within the enterprise has knowledge preconditions, the information flow for the enterprise is completely determined by the structure of the underlying processes. The knowledge preconditions for the processes within the enterprise are achieved by knowledge-producing actions, that is, actions whose effects give the agent knowledge about the state of the world. If the agent does not know the necessary information at the time when the process must occur, a knowledge-producing activity must occur first. In this case, the knowledge-producing actions are analogous to machine setup actions within manufacturing enterprises—if a machine does not have the intended operating conditions when the activity occurs (e.g. a drilling activity requires a particular drill bit), a setup activity must occur first (e.g. the drill bit is changed).

Thus, a process will be integrated with the organization structure of the enterprise if there exist knowledge-producing actions for each process, and there do not exist unnecessary knowledge-producing actions which are not associated with any enterprise process.

4.2.3. Evaluation Processes. The evaluation processes are part of the taxonomy of knowledge-producing actions required by the Business Process Ontology.

Information gathering actions are actions performed by the same agent who performs some business process. They occur when the agent does not have sufficient information at the time at which the business process is intended to occur. Within the Opportunity Management Process, validation processes are performed by gathering information about a customer before committing to achieving a customer's request.

Notification activities are performed by agents who are distinct from the agents which require some information for a business process. Thus, an agent may be notified of some change in the state of the world by another agent, rather than performing the information gathering activities itself.

Condition checking actions are specified with respect to decision making and review activities. All such activities are triggered by some state of the world, but knowledge-producing actions are required if the agent is to know whether or not the condition actually holds.

Information recording activities introduce the notion of an information resource. Rather than performing the necessary information gathering activities, an agent may simply review the information recorded by previous occurrences of the information gathering activities performed by other agents.

4.2.4. Unintended Behaviour. A process will be integrated with the organizational constraints of the enterprise if the agents perform the process as intended in the specification and satisfy all of the associated constraints. This requires that the ontology is able to explicitly represent the difference between intended behaviour and actual behaviour of agents within the enterprise.

The primary intended agent constraint considered within OMP was the set of policies (e.g. the transaction contact must respond to a customer request within 24 hours). Thus the above problem reduces to the problem of determining whether any policies are violated during the process. Further, given that a violation occurs, there must exist an activity within the enterprise which identifies the violation. Such a notification activity is triggered by the occurrence of a violation and causes the violation to become known to an agent.

5. Summary

A rigorous foundation for enterprise design and analysis requires a formal specification of the semantics of enterprise models through the use of ontologies. Any enterprise model must have the expressiveness to capture the sets of constraints satisfied by the agents of the enterprises, the activities that they perform, and the resources required by these activities. The ontologies enable a precise specification of enterprise structure and the application of this structure to the problem of designing an enterprise whose processes are integrated with its organization structure and behaviour. This work not only has firm theoretical foundations, but it has also been validated in applications to business processes within IBM Canada.

Note

1. The complete set of TOVE ontologies and their axiomatization can be found at http://www.eil.utoronto.ca/EIL/tove/toveont/html.

References

Atefi, K. (1997), *Formal Models of Business Process Engineering for Design and Design Validation*, M.A.Sc., Enterprise Integration Laboratory TR-EIL-97-1, Department of Industrial Engineering.

Baid, N. (1995), *Toward an Ontology of Change for Concurrent Engineering*, M.A.Sc. Department of Industrial Engineering, University of Toronto.

Fadel, F., M.S. Fox and M. Gruninger (1994), "A Generic Enterprise Resource Ontology," *Proceedings of the Third Workshop on Enabling Technologies—Infrastructures for Collaborative Enterprises*, West Virginia University.

Fox, M.S. and M. Gruninger (1994), "Ontologies for Enterprise Integration," *Proceedings of the 2nd Conference on Cooperative Information Systems*, Toronto, Ontario.

Fox, M.S. and M. Gruninger (1998), Enterprise Modelling, *AI Magazine*, AAAI Press, Fall 1998, pp. 109–121.

Fox, M.S., M. Barbeceanu and M. Gruninger (1995), "An Organization Ontology for Enterprise Modelling: Preliminary Concepts for Linking Structure and Behaviour," *Computers in Industry*, 29, 123–134.

Genesereth, M. and R. Fikes (1992), *Knowledge Interchange Format Reference Manual*, Report Logic-92-1, Department of Computer Science, Stanford University.

Lin, J., M.S. Fox and T. Bilgic (1996), "A Requirement Ontology for Engineering Design," *Proceedings of 3rd International Conference on Concurrent Engineering*, pp. 343–351. A revised version appears in *Concurrent Engineering*: *Research and Applications*, 4(4), 279–291.

Reiter, R. (1991), "The Frame Problem in the Situation Calculus: A Simple Solution (Sometimes) and a Completeness Result for Goal Regression," *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*. San Diego: Academic Press.

Uschold, M., M. King, S. Moralee and Y. Zorgios (1997), "The Enterprise Ontology," *Knowledge Engineering Review*, 13, 47–76.