A Context-Based Enterprise Ontology

Mauri Leppänen
Department of Computer Science and Information Systems
P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland
mauri@cs.jyu.fi

Abstract

The main purpose of an enterprise ontology is to promote the common understanding between people across different enterprises. It serves also as a communication medium between people and applications, and between different applications. This paper outlines a top-level ontology, called the context-based enterprise ontology, which aims to promote the understanding of the nature, purposes and meanings of things in enterprises with providing basic concepts for conceiving, structuring and representing things within contexts and/or as contexts. The ontology is based on the contextual approach according to which a context involves seven domains: purpose, actor, action, object, facility, location, and time. The concepts in the ontology are defined in English and presented in meta models in a UML-based ontology engineering language.

1. Introduction

Numerous applications are run in enterprises to provide information for, and to enable communication between, various stakeholders, inside and outside the organization. Currently, an increasingly large portion of enterprise knowledge is hold, processed and distributed by applications. Enterprise knowledge is "local knowledge" by its nature, in that its meaning and representation is agreed in relatively small, local contexts. A prerequisite for the successful use of applications is, however, that the common understanding about that knowledge is reached and maintained across the enterprise(s). Especially in modern inter- and intra-organizational applications the need to support the understanding of shared knowledge is crucial [2]. This implies that besides technical interoperability, the enterprises are facing with the challenge of achieving semantic and pragmatic interoperability among the applications.

For human beings to understand what individual things in reality mean they need to know what purposes the things are intended for, by whom, when, and where, how they are related to other things and environment, how they have been emerged, created, and/or evolved, when and where, etc. Shortly, they need to know about contexts where the things appear, have appeared, and/or are to be appeared, and also about the things related to them in those contexts. Considering this, it is understandable that context plays an important role in many disciplines, such as in formal logic, knowledge representation and reasoning, machine learning, pragmatics, computational linguistics, sociolinguistics, organizational theory, sociology, and cognitive psychology. In most of these fields, the notion is used, in particular, to specify, interpret, and infer meanings of things through the knowledge about the contexts they appear.

In the recent years a number of enterprise and business ontologies and frameworks (e.g. [8], [38], [25], [9]) have been proposed. Some of them are generic, whereas the others are aimed at specific business fields (e.g. UNSPC, NAICS, and OntoWeb for e-commerce). In addition, there are several enterprise modeling languages (e.g. IEM, EEML, GRAI/Actigrams). The main purpose of an enterprise ontology is to promote the common understanding between people across different enterprises. It serves also as a communication medium between people and applications, and between different applications. Taking into account the significance that the sharing of meanings has in communication within enterprises as well as experiences got from the use of context in capturing meanings in other disciplines, it is surprising how ignored a contextual view is in current enterprise ontologies. We propose that the semantic and pragmatic interoperability of applications in enterprises should be advanced by the more explicit use of context and other contextual concepts in enterprise ontologies.

Our aim in this study is to present a context-based enterprise ontology. It is a top-level ontology [11], which provides a unified view of the enterprise as an

aggregate of contexts. This ontology can be specialized into task ontologies or domain ontologies to meet special needs of the enterprise, but still maintaining connections of the specialized things to their contexts. The concepts in the context-based enterprise ontology are defined in English and presented in meta models in a UML-based ontology representation language. The UML language has been adopted as the basis because it has a very large and rapidly expanding user community, it is supported by widely adopted engineering tools, and there are positive experiences from the use of UML in presenting ontologies (e.g. [5], [39]). We apply a subset of the concepts of the class diagram.

The article is structured as follows. In Section 2 we will define the notion of context and the contextual approach, and describe the overall structure of the context-based enterprise ontology. In Section 3 we will define the contextual concepts of the ontology and present them in meta models. We will end with the summary and conclusions.

2. Context and Contextual Approach

Based on a large literature review about the notion of context in several disciplines, we conclude that a *context* is a whole, composed of things connected to one another with contextual relationships. A thing gets its meaning through the relationships it has with the other things in that context.

To define a proper set of contextual concepts we draw upon relevant theories about meanings. Based on three topmost layers in the semiotic ladder [36], we identify semantics (e.g. case grammar [7]), pragmatics [22], and the activity theory [6], respectively, to be such theories. In semantics, context appears as a sentence context, in pragmatics as a conversation context, and in the activity theory as an action context.

Anchored on this groundwork and some "contextual" approaches (e.g. [35], [31], [27]), we define seven domains, which serve concepts for specifying and interpreting contextual phenomena. These contextual domains are: purpose, actor, action, object, facility, location, and time (Figure 1). Structuring the concepts within and between these domains is guided by the following scheme, called the seven S's scheme: *For **S**ome* purpose, ***S**omebody* does ***S**omething* for ***S**omeone*, with ***S**ome* means, ***S**ometimes* and ***S**omewhere*.

We define the *contextual approach* to be the approach according to which individual things are seen to play certain contextual roles in a context and/or to be contexts themselves. Following this approach, we define an *enterprise* to be an aggregate of contexts that are composed of people, information and technologies,

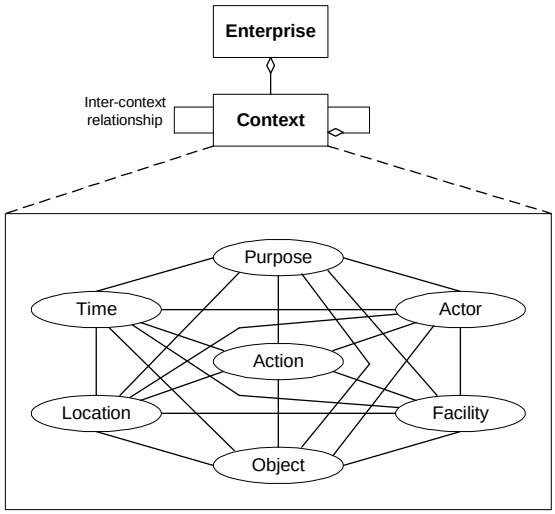

Figure 1. An overall structure of the context-based enterprise ontology

performing functions in a defined organizational structure, for agreed purposes, and responding to events, both internal and external, and needs of stakeholders. The contexts can be decomposed into more elementary contexts, and they are related to one another with inter-context relationships.

An ontology is an explicit specification of a conceptualization of some part of reality that is of interest [10]. The *context-based enterprise ontology* is an ontology which aims to promote the understanding of the nature, purposes, and meanings of the things in the enterprise with providing concepts and constructs for conceiving, structuring, and representing things within contexts, and/or as contexts. The ontology is intended to assist the acquisition, representation, and manipulation of enterprise knowledge via the provision of a consistent core of basic concepts and constructs.

In the next section we will first define the contextual domains and the most essential concepts within them. Due to the limitation of space, the location and time domains are excluded. In addition, we will shortly present relationships between the domains.

3. Contextual Domains

3.1 Purpose Domain

The *purpose domain* embraces all those concepts and constructs that refer to goals, motives, or intentions of someone or something (Figure 2). The concepts are also used to express reasons for which something exists or is done, made, used, etc. We use *purpose* as the general term in this domain.

A *goal* (of e.g. an actor or action) means a desired state of affairs ([25], [19]). It can also be related to an object, a facility, a location or a time (system), meaning the purpose, which they are aimed at. A *reason* is a basis or cause for some action, fact, event etc. [40]. It can be a requirement, a problem, a strength/weakness, or an opportunity/a threat. Between a goal and a reason there is the *dueTo relationship*, meaning that a reason gives an explanation, a justification or a basis for setting a goal.

We can specialize the goals based on their lifespan. *Strategic goals* are kinds of missions, answering questions such as "What is the direction of an enterprise in the future". Their spans are generally 5 – 10 years. *Tactic goals* show how to attain strategic goals. *Operative goals* are generally determined as concrete requirements that are to be fulfilled by a specified point of time. The goals can also be categorized based on whether it is possible to define clear-cut criteria for the assessment of the fulfillment of goals. *Hard goals* have pre-specified criteria, and *soft goals* have not [23].

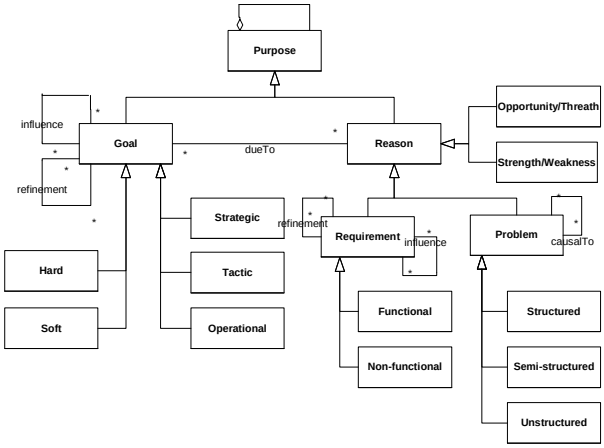

Figure 2. Purpose domain

Requirements mean something that are necessary and needed. They are statements about the future [28]. Actually, the goals and the requirements are two sides of a coin: some of the stated requirements can be accepted to be goals to which actors want to commit. A *functional requirement* can be achieved by performing a sequence of operations [20]. A *non-functional requirement* is defined in terms of constraints, to qualify the functional requirement related to it.

Instead of directly referring to a desirable state, a purpose can also be expressed through an indirect reference to problems that should be solved. A *problem* is the distance or a mismatch between the prevailing state and the state reflected by the goal [15]. To reach the goal, the distance should be eliminated or at least reduced. Associating the problems to the goals expresses reasons, or rationale, for decisions or actions towards the goals [30]. The problems are commonly divided into structured, semi-structured and unstructured problems [33]. *Structured problems* are those that are routine, and can be solved using standard solution techniques. *Semi-structured* and *unstructured problems* do not usually fit a standard mold, and are generally solved by examining different scenarios, and asking "what if" type questions.

Other expressions for the reasons, of not so concrete kind, are strengths, weaknesses, opportunities and threats related to something for which goals are set (cf. SWOT-analysis, e.g. [16]). *Strength* means something in which one is good, something that is regarded as an advantage and thus increasing the possibilities to gain something better. *Weakness* means something in which one is poor, something that could or should be improved or avoided. *Opportunity* is a situation or condition favorable for attainment of a goal [40]. *Threat* is a situation or condition that is a risk for attainment of a goal.

A general goal is refined into more concrete ones. The *refinement relationship* between the goals establishes goal hierarchies, in which a goal can be reached when the goals below it (so-called sub-goals) in the hierarchy are fulfilled (cf. [18]). The *influence relationship* indicates that the achievement of a goal has some influence, positive or negative, on the achievement of another goal (cf. [25], [18]).

As the goals and the requirements are two sides of a coin, the relationships between the requirements are similar to those between the goals. Consequently, a requirement can influence on another requirement, and a requirement can be a refinement of another requirement. The relationships between the problems manifest causality. The *causalTo relationship* between two problems means that the appearance of one problem is at least a partial reason for the occurrence of the other problem.

3.2 Actor Domain

The *actor domain* consists of all those concepts and constructs that refer to human and active parts in a context (Figure 3). Actors perform, own, communicate, borrow, send, receive etc. objects in the contexts. They are responsible for and/or responsive to triggering and causing changes in the states of objects in the same context, or in other contexts. We consider it important, from the philosophical viewpoint, to distinguish human actors from non-human actors, which are here regarded as tools (see Section 3.5).

An *actor* is a human actor or an administrative actor. A *human actor* is an individual person or a group of persons. A *person* is a human being, characterized by

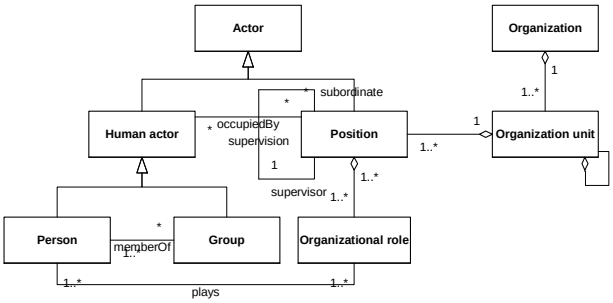

Figure 3. Actor domain

his/her desires, intentions, social relationships, and behavior patterns conditioned by his/her culture (cf. [3], [29]). A person may be a member of none or several *groups.* An administrative actor is a position or a set of positions. A *position* is a post of employment *occupied by* zero or many human actors. For each position, specific qualifications in terms of skills, demands on education and experience, etc. are specified.

An *organizational role*, shortly a role, is a collection of responsibilities, stipulated in an operational or structural manner. In the former case, a role is composed of tasks that a human actor occupying the position with that role has to perform. In the latter case, a role is charged with responsibilities for some objects. A role can be played by many persons, through or without the position(s) they hold.

The *supervision relationship* involves two positions in which one is a supervisor to another that is called a subordinate. A supervisor position has responsibility and authority to make decisions upon the positions subordinate to it, and those occupying the subordinate positions have responsibility for reporting on one's work and results to those occupying the supervisor position.

An *organization* is an administrative arrangement or structure established for some purposes, manifesting the division of labor into actions and the coordination of actions to accomplish the work. It can be permanent and formal, established with immutable regulations, procedures and rules. Or it may be temporally set up, like a project organization, for specific and often short-range purposes. An *organizational unit* is composed of positions with the established supervision relationships. An organization consists of organizational units.

3.3 Action Domain

The *action domain* comprises all those concepts and constructs that refer to deeds or events in a context (Figure 4). We use *action* as the generic concept to refer to things belonging to the action domain. Actions can be autonomous or cooperative. They can mean highly abstract work like studies in mathematics, or at the other extreme, physical execution of a step-by-step procedure with detailed routines.

There are a large number of action structures, which an action is a part of. We distinguish between the decomposition structure, the control structure, the temporal structure and the management – execution (Mgmt-Exec) structure.

In the *decomposition structure*, actions are divided into sub-actions, these further into sub-sub-actions, etc. Sub-actions may be functions, activities, tasks, operations, etc. Decomposition aims at reaching the level of elementary actions, where it is not possible or necessary to further decompose. The *control structure* indicates the way in which the actions are logically related to each other and the order in which they are to be executed. The control structures are: sequence, selection, and iteration. The *sequence relationship* between two actions act_1 and act_2 means that after selecting the action act_1 the action act_2 is next to be selected. The *selection relationship* means that after selecting the action act_1 there is a set of alternative actions act_2,.., act_n from which one action (or a certain number of actions) is to be selected. The *iteration relationship* means that after selecting the action act_1 the same action is selected once more. The selection is repeated until the stated conditions become true. The *temporal structure* is like the control structure but with temporal conditions and events.

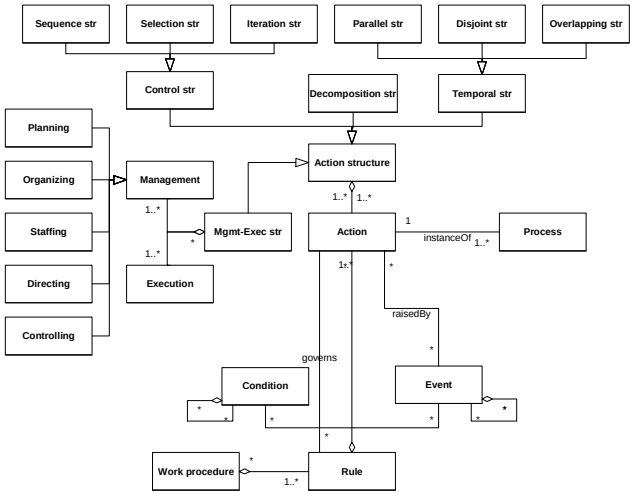

Figure 4. Action domain

The temporal structures are specified using temporal constructs, such as during, starts, finishes, before, overlaps, meets, and equal. Constructs are used to specify relationships between starting and/or ending events, or between durations of actions. With these

constructs, overlapping, parallel, disjoint (non-parallel) and overlapping executions of actions can be distinguished. Two actions are said to be *overlapping* if the durations of their executions overlap. The actions are (strictly) *parallel* if the durations are equal or the duration of one action is included in the duration of the other action. Two actions are said to be *disjoint* if their durations do not overlap.

The *management – execution structure* is composed of one or more management actions and those execution actions that implement prescriptions provided by the management actions (e.g. [26], [41], [14]). *Management actions* mean the *planning, organizing, staffing, directing and controlling* of execution actions, in order to ensure the achievement of goals and constraints (cf. [4], [34], [37]). The purpose of *execution actions* is to implement the prescriptions with the given resources.

The action structures are orthogonal to one another. This makes it easy to specialize the defined structures, and extend them with new ones, e.g. with the dichotomy of material and social actions (cf. speech acts [32]). The action structures are enforced by rules. A *rule* is a principle or regulation governing a conduct, action, procedure, arrangement, etc [40]. It is composed of four parts [12], event, condition, thenAction, and elseAction, structured in the ECAA structure. An *event* is an instantaneous happening in the context, with no duration. A *condition* is a prerequisite for triggering an action. A *thenAction* is an action that is done when the event occurs and if the condition is true. An *elseAction* is an action that is done when the event occurs but the condition is not true. An aggregate of related rules constitutes a *work procedure* (cf. [14]), which prescribes how the course of action should proceed. Depending on the knowledge of, and a variety of, actions, work procedures may be defined at different levels of detail [13]. An instance of an action is a *process*.

3.4 Object Domain

The *object domain* contains all those concepts and constructs that refer to something, which an action is directed to (Figure 5). It can be a message, a decision, an argumentation, a list of problems, a program code, a workstation, etc. In general, an object can be a conception in a human mind, data represented in some carrier, or physical material (cf. the semiotic realms). We use *object* as the generic term to signify any concept in the object domain.

Based on the nature of the objects we can distinguish between material objects and informational objects. *Material objects* do not carry or present any information, whereas *informational objects* do. For us,

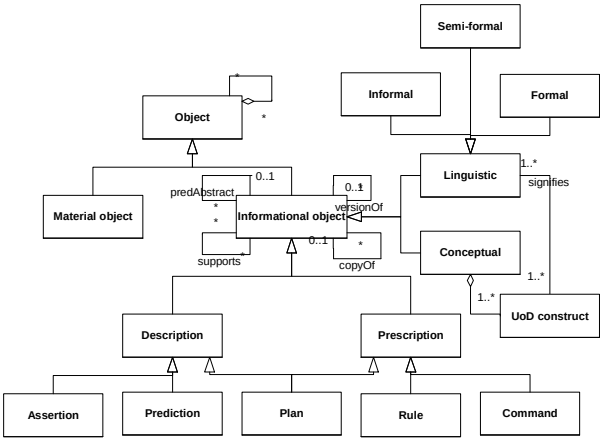

Figure 5. Object domain

objects of special interest are in the form of data or information. We call them *linguistic objects* and *conceptual objects*, respectively. Linguistic objects can be *formal*, *semi-formal* or *informal.*

Informational objects can be classified based on the intentions by which they are provided and used (e.g. [36], [32], [21]). Informational objects can be descriptive or prescriptive. A descriptive object, called a *description*, is a representation of information about a slice of reality. An informational object can be descriptive in various ways. An *assertion* is a description, which asserts that a certain state has existed or exists, or a certain event has occurred or occurs. A *prediction* is a description of a future possible world with the assertion that the course of events in the actual world will eventually lead to this state (cf. [21]). A prescriptive object, called a *prescription*, is a representation of the established practice or an authoritative regulation for action. It is information that says what must or ought to be done. A prescription with at least two parts ((E or C) and A) of the ECAA structure is called a *rule*. A prescription with neither an event part nor a condition part is called a *command*. A *plan* is a description about what is intended. It can also be regarded as a kind of prediction, which is augmented with intentions of action. It is assumed that the future possible world described in the plan would not normally come out, except for the intended actions (cf. [21]).

An object is often produced gradually through several iterations. The *versionOf relationship* holds between two objects obj_1 and obj_2, if properties of, and experience from, the object obj_1 have influenced the creation of another object obj_2 intended for the same purposes (cf. [17]). We may also have several copies from an object. The *copyOf relationship* holds between two objects, the original object and a copy object,

which are exactly, or to an acceptable extent, similar. The *supports relationship* involves two informational objects, obj_1 and obj_2, such that the information "carried" by the object obj_1 is needed to produce the object obj_2. The *predAbstract relationship* between two informational objects means that one object is more abstract that the other object in terms of predicate abstraction and both of the objects signify the same thing(s) in reality. The *signifies relationship* defines the conceptual meaning of a linguistic object in terms of UoD constructs, which the object signifies. The *UoD construct* means any conceptual construct. The *partOf relationship* means that an object is composed of two or more other objects.

3.5 Facility Domain

The *facility domain* contains all those concepts and constructs that refer to the means by which something can be accomplished, i.e. something, which makes an action possible, more efficient or effective (Figure 6). We distinguish between two kinds of *facilities*, tools and resources.

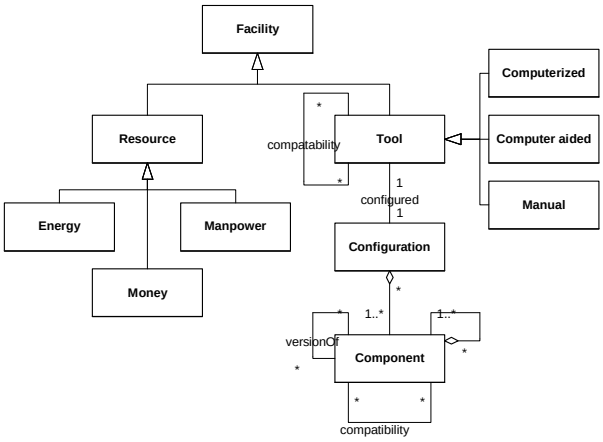

Figure 6. Facility domain

A *tool* is a thing that is designed, built, installed, etc. to serve in a specific action affording a convenience, efficiency or effectiveness. A tool may be a simple and concrete instrument held in hand and used for cutting or hitting. Or, it may be a highly complicated computer system supporting an engineer in his/her controlling a nuclear power station. Tools can be *manual, computer aided,* or *computerized.* A *resource* is a kind of source of supply, support, or aid. It can be money, energy, capital, goods, manpower, etc. [1]. The resources are not interesting in terms of pieces, but rather in terms of amount. When a resource is used, it is consumed, and when consuming, the amount of the resource diminishes. Thus, a resource is a thing, about which the main concern is how much it is available (cf. [24].

There are a great number of relationships between the concepts within the facility domain, representing e.g. functional and structural connections. We consider only some of them. For being operative and useful, tools should be compatible**.** Two tools are *compatible* if their interfaces are structurally and functionally interoperable. Tools are composed of one or more components that develop through consecutive *versions*. Only some versions of a component are compatible with certain versions of the other components. A *configuration* is a whole that is composed of the components of compatible versions.

3.6. Inter-Domain Relationships

Until now we have defined only those contextual relationships which associate concepts within the same contextual domain. There is, however, a large set of contextual relationships that relate concepts in different domains. For example, an actor carries out an action, an object is an input to an action, and a facility is situated in a location. We call these *inter-domain relationships.* Figure 7 presents an overview of inter-domain relationships. The space is divided into seven sub-areas corresponding to the seven contextual domains. In each of the sub-areas we present the concerned generic concepts to be related with the inter-domain relationships. It goes beyond the space available to define the relationships here.

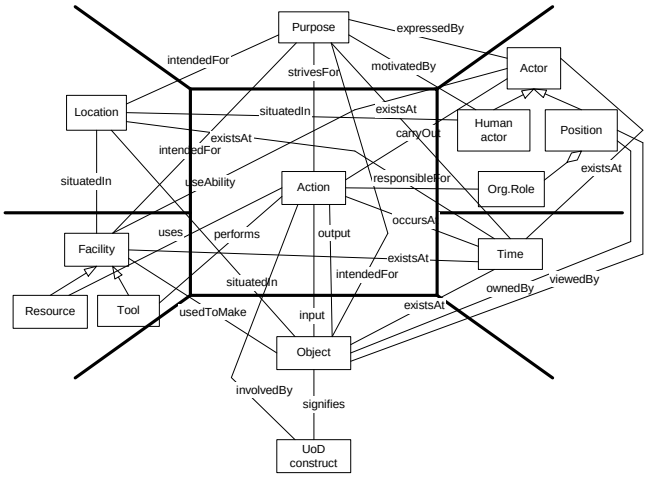

Figure 7. Overview of inter-domain relationships

In addition to the binary inter-domain relationships, there are multiple n-ary relationships. With these, together with composing binary inter-domain relationships, it is possible to specify things in the

enterprise in a way that reveals their contextual meanings. An example of this kind of specification is: the customer c places the order o for the product p at time t, based on the offer o from the enterprise e, owned by the partners $\{p_1,...p_n\}$, to be delivered by a truck tr to the address a by the date d. It depends on the situation at hand which contextual domains and concepts are seen to be relevant to be included in the specification.

4. Summary and Conclusions

In this article we have presented the context-based enterprise ontology to promote the understanding of the nature, purposes, and meanings of things about which information is stored and processed in, and transmitted between, various applications in enterprises. This ontology, grounded upon theories, such as case grammar, pragmatics and activity theory, guides a conceptualization of the structure and behavior of the enterprise through considering things as contexts, and/or as parts thereof.

Although our ontology, as having been derived from relevant theories, inherently embodies essential contextual concepts, it is just a top ontology. At this stage, it can be deployed as a frame to analyze and compare other enterprise ontologies in terms of their contextuality. Later, our ontology should be specialized into a task ontology, or a domain ontology, for the needs of a specific business task or field. Experiments made on such kinds of specializations and comparisons of their outcomes with current enterprise ontologies indicate that existing enterprise ontologies lack many essential contextual concepts and constructs and some of the conceptual constructs in them should be reengineered, in order to enable the recognition, representation and derivation of meanings in enterprise knowledge. Unfortunately, it goes beyond the space available to consider this further here. Continuing our top down approach to ontology engineering, we will next focus on a more systematic derivation of specialized concepts and constructs, and use them in empirical studies on semantic and pragmatic interoperability of enterprise applications. In this phase, we aim also to validate our ontology.

References:

[1] O. Barros, "Modeling and evaluation of alternatives in information systems", *Information Systems*, Vol. 16, No. 5, 1991, pp. 537-558.

[2] D. Bianchini, V. De Antonellis and M. Melchiori, "Ontology-based semantic infrastructure for service interoperability for interorganizational applications", In M. Missikoff (Ed.) *Proc. of the Open InterOp Workshop on Enterprise Modelling and Ontologies for Interoperability*, 7-8 June 2004, Riga, Latvia, 2004.

[3] Bratman M., *Intentions, plans, and practical reason*, Harward University Press, Cambridge, 1987.

[4] Cleland D. and W. King, *Management: a systems approach*, McGraw-Hill, New York, 1972.

[5] S. Cranefield and M. Purvis, "UML as an ontology modeling language", In Proc. of the Workshop on *Intelligent Information Integration*, held in conjunction with the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99), 1999.

[6] Engeström Y., *Learning by expanding: an activity theoretical approach to developmental research*, Orienta-Konsultit, Helsinki, 1987.

[7] C. Fillmore, "The case for case", In E. Bach and R. T. Harms (Eds.) *Universals in Linguistic Theory*, Holt, Rinehart and Winston, New York, 1968, pp.1-88.

[8] M. Fox, "The TOVE Project: A common-sense model of the enterprise", In F. Belli and F. Radermacher (Eds.) *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, LNAI 604, Springer-Verlag, Berlin 1992, pp. 25-34.

[9] G. Geert and W. McCarthy, "The ontological foundations of REA enterprise information systems", 2000, online: htt://www.msu.edu/user/mccarh4/rea-ontology/

[10] T. Gruber, "A translation approach to portable ontology specification", *Knowledge Acquisition*, Vol. 5, No. 2, 1993, pp. 119-220.

[11] N. Guarino, "Formal ontology and information systems", In N. Guarino (Ed.) Proc. of Conf. on *Formal Ontology in Information Systems (FOIS'98)*, IOS Press, Amsterdam, 1998, pp. 3-15.

[12] H. Herbst, "A meta-model for business rules in systems analysis", In J. Iivari, K. Lyytinen & M. Rossi (Eds.) *Advanced Information Systems Engineering*, LNCS 932, Springer, Berlin, 1995, pp. 186-199.

[13] Hoc J.-M., *Cognitive psychology of planning*, Academic Press, London, 1988.

[14] J. Iivari, "Levels of abstraction as a conceptual framework for an information system", In E. Falkenberg and P. Lindgren (Eds.) *Information System Concepts: An In-Depth Analysis*, North–Holland, Amsterdam, 1989, pp. 323-352.

[15] Jayaratna N., *Understanding and evaluating methodologies: NIMSAD – a systemic framework*, McGraw-Hill, London, 1994.

[216] Johnson G., K. Scholes and R.W. Sexty, *Exploring strategic management*, Prentice-Hall, Englewood Cliffs, 1989.

[17] R. Katz, "Toward a unified framework for version modeling in engineering databases", *ACM Surveys*, Vol. 22, No. 4, 1990, pp. 375-408.

[18] V. Kavakli and P. Loucopoulos, "Goal-driven business process analysis application in electricity deregulation", *Information Systems*, Vol. 24, No. 3, 1999, pp. 187-207.

[19] M. Koubarakis and D. Plexousakis, "A formal model for business process modeling and design", In B. Wangler and L. Bergman (Eds.) Proc. of 12th Int. Conf. on *Advanced Information Systems Engineering (CAiSE 2000)*, Springer-Verlag, Berlin, 2000, pp. 142-156.

[20] J. Lee, N.-L. Xue and J.-Y. Kuo, "Structuring requirement specifications with goals", *Information and Software Technology*, Vol. 43, No. 2, 2001, pp. 121-135.

[21] R. Lee, "Epistemological aspects of knowledge-based decision support systems", In H. Sol (Ed.) Proc. of Int. Conf. on *Processes and Tools for Decision Support Systems*, North-Holland, Amsterdam, 1983, pp. 25-36.

[22] Levinson S., *Pragmatics*, Cambridge University Press, London,1983.

[23] C.-Y. Lin, C.-Y. and C.-S. Ho, "Generating domain-specific methodical knowledge for requirements analysis based on methodology ontology", *Information Sciences*, Vol. 14, No. 1-4, 1999, pp. 127-164.

[24] L. Liu and E. Yu, "Designing web-based systems in social context: a goal and scenario based approach", In A. Banks Pidduck, J. Mylopoulos, C. Woo and M. Tamer Ozsu (Eds.) Proc. of 14th Int. Conf. on *Advanced Information Systems Engineering (CAiSE'2002)*, LNCS 2348, Springer-Verlag, Berlin, 2002, pp. 37-51.

[25] P. Loucopoulos, V. Kavakli, N. Prekas, C. Rolland, G. Grosz and S. Nurcan,. Using the EKD approach: the modelling component. ELEKTRA – Project No. 22927, ESPRIT Programme 7.1, 1998.

[26] Mesarovic M., D. Macko, and Y. Takahara, *Theory of hierarchical, multilevel, systems*, Academic Press, New York, 1970.

[27] H. Myrhaug, "Towards life-long and personal context spaces", In Proc. of Workshop on *User Modelling for Context-Aware Applications*, 2001.

[28] NATURE Team, "Defining visions in context: models, processes and tools for requirements engineering", *Information Systems*, Vol. 21, No. 6, 1996, pp. 515-547.

[29] L. Padgham and G. Taylor, "A system for modeling agents having emotion and personality", In L. Cavedon, A. Rao & W. Wobcke (Eds.) *Intelligent Agent Systems*, LNAI 1209, Springer-Verlag, Berlin, 1997, pp. 59-71.

[30] R. Ramesh and A. Whinston, "Claims, arguments, and decisions: formalism for representation, gaming, and coordination", *Information Systems Research*, Vol. 5, No. 3, 1994, pp. 294-325.

[31] C. Rolland, C. Souveyet and M. Moreno, "An approach for defining ways-of-working", *Information Systems*, Vol. 20, No. 4, 1995, pp. 337-359.

[32] Searle J. and D. Vanderveken, *Foundations of illocutionary logic*, Cambridge University Press, New York, 1985.

[33] Simon H., *The new science of management decisions*, Harper & Row, New York, 1960.

[34] Sisk H., *Management and organization*, South Western Pub. Co., International Business and Management Series, Cincinnati, 1973.

[35] J. Sowa and J. Zachman, "Extending and formalizing the framework for information system architecture", *IBM Systems Journal*, Vol. 31, No. 3, 1992, pp. 590-616.

[36] R. Stamper, "Information science for systems analysis", In E. Mumford and H. Sackman (Eds.) *Human Choice and Computers*, North-Holland, Amsterdam, 1975, pp. 107-120.

[37] R. Thayer, "Software engineering project management – a top-down view", In R. Thayer (Ed.) *Tutorial: Software Engineering Project Management*, IEEE Computer Society Press, 1987, pp. 15-56.

[38] M. Uschold, M. King, S. Moralee and Y. Zorgios, "The Enterprise Ontology", *The Knowledge Engineer Review*, Vol. 13, No. 1, 1998, pp. 31-89.

[39] X. Wang and C. Chan, "Ontology modeling using UML", In Y. Wang, S. Patel and R. Johnston (Eds.) Proc. of the 7th Int. Conf. on *Object-Oriented Information Systems (OOIS'2001)*, Springer-Verlag, Berlin, 2001, pp. 59-70.

[40] Webster, *Webster's Encyclopedic Unabridged Dictionary of the English Language*, Gramercy Books, New York, 1989.

[41] Weick K.E., *Sensemaking in organizations*, Sage Publications, California, 1995.