The Role of Ontology in Building of Knowledge Systems for Industrial Applications

Baltazár Frankovič, Ivana Budinská

Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia, frankovic@savba.sk

Abstract: [1]Practical aspects of ontological engineering in building knowledge systems for the application domain of production systems as a whole and two other applications are presented in this paper. The Section 2 presents several ontology development methodologies. Section 3 is devoted to ontology representation languages and ontology editors. The application of the methodology for building knowledge system is discussed in Section 4. Section 5 presents ontology utilisation for industrial applications.

Keywords: Ontology, manufacturing network

1 Introduction

The knowledge system based on the ontology has a very important role in building of a manufacturing network (MN). The most important problem in the performance of manufacturing system or manufacturing network is:

- dimensioning the system
- planning the production
- scheduling the production
- taking account the customer requirement
- taking account random events and maintenance
- specifying internal rules of production.

The new manufacturing contexts require a lot of flexibility [1]. Most of the products under continuous revision, trying to fulfill the changing consumers needs. For many products, managers vary lot sizes in order to respond to demand in a just in time manner while keeping the inventories as low as possible. The lots

[1] The work was supported by APVT and VEGA agencies under projects No APVT 51-011602, 51 024604 and VEGA 2/4148/24.

sizes can often vary from few units to a few hundreds of units for the same product and within production periods. Manufacturing and delivery leads times are continuously becoming shorter. Competition is coming from all over the world.

In the manufacturing facility operating under agile or mass customizing paradigms, a network approach appears to be the most efficient and adaptable structure.

This type of manufacturing facility configuration can add or delete resources and links, on an *as required* basis. In such a network nodes may be, for example, workstation, equipments, special crews, subcontractors, handling equipment or any specific resources required to complete a particular task. These resources can be combined together for very short period of time, a few hours or days, or on a near permanent basis. In a given network, many workstations, manufactories, enterprises can have the same equivalent or different technological capabilities. These equipments can operate under a similar cost structure or have different operating and marginal costs.

Given the network and processors flexibility, hundreds or even thousand of potential combinations of workstations, manufactories, enterprises are conceivable to complete a particular task. It is not a trivial role to select the most efficient and cheapest combination. The situation is even more complex most of the time, many task or products can be realized simultaneously in a network, using sequentially the same processors, with various lot sizes and routings. To efficiently utilize the manufacturing resources, it becomes evident that it is imperative to develop new adequate manufacturing systems design, planning and control tools. In the next part we try to describe some problems of MN by the **Virtual Manufacturing Networks.**

Virtual manufacturing network allow the design of temporary networks, minimizing operating costs. Theses networks are designed to respect two main subsets. The first subset is representing classical constraints; it ensures basic flows and capacity feasibility in the network. The second subset is a deterministic approximation mechanism for ensuring that the resulting flow and work variables are not to be prescriptive deterministic static work-flow assignments but rather descriptive estimates of forth coming real work-flow patterns.

The operating conditions considered the following outputs:

- the expected operating cost over the planning horizon
- the expected work pattern for the various workstation, manufactories, enterprises over the entire network
- the expected product flow pattern among the workstation. manufactories, enterprises
- the expected manufacturing network for each product.

The information contained in the three last outputs is synthesized into the virtual manufacturing network concept.

Virtual manufacturing network may be computed for each product and for given production period, one production day, for example. They have to be recomputed for each production period based on the new demand and manufacturing conditions.

To ensure the feasibility and to assign tasks to workstations, manufactories, enterprises in a time based manner, it is necessary to develop new algorithms and approaches created on the basis of artificial intelligence.

To generate the virtual manufacturing network many variables and constraints have to be considered. The virtual manufacturing network using the intelligent algorithms and agent technologies in the form of agent coalition systems can be a good tool for finding of the best production conditions, which ensure the minimum of the total manufacturing handling, inventory and costs.

2 Ontology Development Methodology

Creating ontology is not a trivial problem. It requires not only the skills in information technologies but also a great knowledge in the modelled domain. To easy the process of ontology creation a couple of methods were suggested. The basic principles for building ontology may be derived from the CommonKADS methodology [2], which deals with the common principles of knowledge systems development. CommonKADS methodology was developed within a series of international research and application projects. The process of knowledge system development is structured in a couple of models that have to be created. On the "context" level of abstraction the three models are suggested: Organizational model, Task model and agent's model. The organizational model describes the organization with the aim to discover the problems and opportunities of knowledge management. The task model represents task that are performed within the organization. The task is anything that has to be executed by an agent. The agent model describes all agents – executors of tasks – their roles, competencies, capabilities, and limitations. Above the contextual level lay a conceptual level that covers Communication model and Knowledge model. The models are derived from the three models in the conceptual level. The knowledge model describes knowledge that is required to perform the tasks. The communication model figures communicative transactions between agents that perform the task. Finally Design model is an artifact that describes the structure of a knowledge system to be created.

Overview of Ontology Development Methodologies

Ontology development methodologies help creating ontologies in various domain oriented application. Several methodologies have been developed in order to formalize creating ontologies for industrial or other applications. Although ontology development methodologies are not mature enough, they can be helpful in developing ontology based knowledge systems. The overview of some methodologies are given e.g. in [3], [4], or [5].

METHONTOLOGY

The Methontology [7] has been developed for Software Life Cycle Processes. It supports project management processes (contains guidelines for planning, project control, quality control, etc.), ontology development processes (contains guidelines for use of ontology, conceptualization of domain, formalization of ontology, implementation, etc.), and support activities (guidelines for knowledge acquisition, evaluation, ontology integration, documentation, version management, etc.).

TOVE Methodology

The TOVE methodology [6] was developed at Toronto University in order to help modelling of enterprise processes. The methodology goes from informal definitions to the formal competency questions. The ontology must provide vocabulary to answer these questions. First the informal competency questions have to be answered and the basic terms from these answers are extracted. Using vocabulary the informal competency questions are formalized and the ontology has to be evaluated if it is complete.

On-To Knowledge methodology [8] was developed on the basis of KADS methodology. It also uses a method of competency questions [9]. On-To-Knowledge methodology uses a two-loop architecture, which is composed of knowledge processes and knowledge metaprocesses. Knowledge metaprocesses describes building ontology in 5 basic steps (with 13 sub-steps): Feasibility study, Kick-off, Refinement, Evaluation, Application and evaluation.

Methodology by Ushold and King

The methodology by Ushold and King was developed within the Enterprise project and was used by Enterprise Ontology [18] creation. However the methodology is general and may be used in other domain. The skeleton of Usholds and Kings' methodology contains four basic steps: Identification the purpose of ontology building, building the ontology, evaluation and documentation. Ushold and King' methodology assume the informal ontology development and then the formalization of the informal ontology by any of formal ontological languages. The procedure of informal ontology development include collection of concepts by brainstorming, clustering of the collected concepts, and refinement of the concept set by investigating what concepts are basic, what are generic, or specific, what relations are among them. The name of concepts has to be determined. Each concept has to be named by an original name that has only one meaning in the ontology. The meaning of the names has to be defined for each concept. The importance of informal ontology that is comprehensible for many people are the crucial idea of this methodology. Methodology by Ushold and King belongs to the most formalized methodologies and can be successfully used in many domain applications.

3 Ontology Languages and Tools

Informal ontology has to be represented by one of the formal ontology languages in order to build computer processed ontology that is only usable in knowledge management systems. Usually ontology development methodology has its own tool to support ontology and instances in formal ontology representation language. The brief description of some most used ontology languages and tools is given in this Section.

Ontolingua

Ontolingua [11] is originally an interlingua for ontology representation and sharing developed by KSL (Knowledge Systems Lab) at Stanford University. It is designed by adding frame-like representation and translation functionalities to KIF (Knowledge Interchange Format) which is a logic-based Interlingua for knowledge representation. It can translate from and to some description logics languages. Ontolingua itself does not have inference functionality. It has currently developed into a development environment which provides a set of ontology development functions (browse, create, edit, modify and use ontology) and a library of modular and reusable ontologies.

RDF(S)

RDF(Resource Description Framework) is a framework for metadata description developed by W3C [14]. It defines the triplet <object, attribute, value>, in which object is called resource and can be representing by a web page, url address, etc. A triplet itself can be an object and a value. Value can be a string or resource. Attrributes represents links between objects and values. RDF model is a base for creating a semantic network. RDF has an XML-based syntax (called serialization). But, RDF is different from such a language in that it is a data representation model rather than a language and that the XML's data model is the nesting structure of information and the frame-like model with slots.

OWL (DAML+OIL)

An OWL – Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics [15]. The application of the OWL format for ontology for the agent system is relatively new. One advantage of owl ontology is the availability of tools that can reason about it. Tools provide generic support that is not specific to the particular subject domain. Constructing ontology in owl enables to benefit from third party tools based on the formal properties of the OWL language.

OntoEdit

OntoEdit [10] is a professional tool that helps to create ontology based on On-To-Knowledge methodology and CommonKADS. OntoEdit contains inference machine based on the F-Logic. It plays crucial role in the evaluation process. Opposite to the description logic, F-Logic can express arbitrary powerful rules which quantify over the set of classes.

Protégé

Protégé [11] is a powerful tool for building and creating domain ontology. It supports some of formal ontology languages as RDF, and OWL, contains customizable user interface, and has powerful plug-in architecture, that enables integration with other applications.

4 Examples of Some Possible Applications

Two examples, where the possible applications of the ontology tools in building of knowledge system are introduced in this part.

4.1 Example of an Ontology- Centric Multi-agent System for Coordinating Medical Responses to Large-scale Disasters

By Peter Bloodsworth and Sue Greenwood was developed a multi-agent system with acronym COSMOA to support the decision making process during the medical response to a large-scale incident [16]. The system simulates the monitoring of a range of news feeds in addition to emergency service reports of incidents in order to determine that an incident has taken place, and the nature of the incident. The design of COSMOA makes full use of ontologies in delivering its many features. Ontologies are used within COSMOA to collect, integrate and reason on heterogeneous data. By placing ontologies at the heart of the multi-agent system and making the maximum of use them was created a system in which agent behaviors and internal representation are abstracted from the coding. This approach places rules, heuristics and statistical attributes that define agent behavior in an ontology layer. Each agent in the system uses this layer, in addition to instances, to form a knowledge base defining its behavior. The nature of the ontology layer depends on the problem domain being considered and can range from a single ontology to a group of many generic and domain specific ontologies.

The multi-agent architecture for COSMOA (Figure 1) needed to be created in such a way that it facilitated ontology-centric design. The main objective was to limit the number of types of agents that are required.

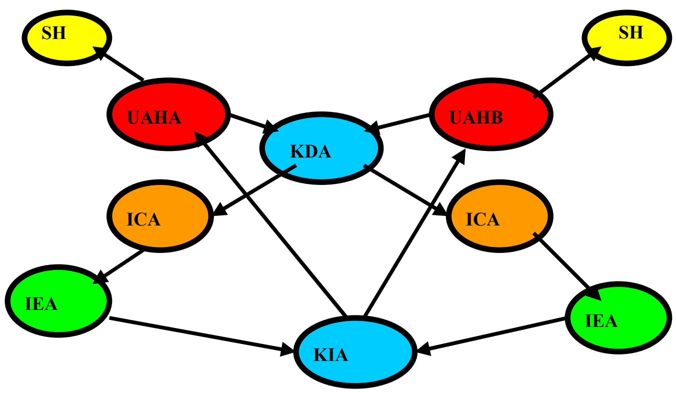

Figure 1

The COSMOA architecture. SH- Scheduler Agent; UAHA- User Agent Hospital A; UAHB- User Agent Hospital B; KDA- Knowledge Directory Agent; ICA- Information Collection Agent; IEA- Information Extraction Agent; KIA- Knowledge Integration Agent

The user and knowledge integration agents use a rule- engine to decide the impact that various facts have on their environment. The knowledge integration agent uses rules to determine the effect that a fact has on the world in general and then passes these environmental changes on the user agents. The knowledge directory agent keeps track of the information sources that are available to the system and the access methods required. In order to do this it uses the ontology layer to structure the data and to communicate it to the information collection agents. The ontology layer also defines and structures the access methods and the resources that are available. The raw information is passed to the information extraction agent, which then extracts the necessary data. The scheduler will make use of a generic scheduling ontology located within the ontology layer. A domain specific ontology allows the generic scheduling methods to be applied to a specific problem domain. This architecture can be applied in the domain of disaster planning.

4.2 Example of Ontologies Application in Business Environment

An ontology management system represents a powerful tool to create common and shareable knowledge repositories concerning a business application domain. In [17] is presented SymOntoX (Symbolic Ontology XML-based management system) a software prototype for the management of domain ontologies. SymOntoX offers a few native modeling notions (referred to as meta-concepts), such as Business Process, business Objects, and Business Actor that help the enterprise experts to better categorize the identified concepts. It denotes that in this perspective, its ontology model, referred to as OPAL (Object, Process, and Actor modeling Language), which according the opinion [13] is more focused to business modeling then other existing system, such as Protégé [11], OntoEdit [10], KSL Ontology Server [12].

The OPAL acronym, in SymOntoX an ontology is organized to three primary meta-concepts:

Actor_kind – aimed at modeling any relevant entity of the domain that is able to activate or peform a process (e.g.*Tourist, Travel Agency);*

Process_kind – aimed at modeling an activity that is performed by an actor to achive a given goal (e.g. *Making_a_reservation*);

Object_kind – aimed at modeling a passive entity, on which a process operates (e.g. ,*Hotel, Flight*), typically to modify its state.

A ontology represents several aspects of the business domain. In principle, such aspects can be modeled as properties of primary concepts (as instance of primary meta-concepts). The list of such meta-concepts may be as follows:

Goal- a desired state of the affar;

State- a characteristic pattern of values *(e.g Flight_full);*

Rule- an expression that is aimed at restarting the possible values of an instance of a concept or that allow to derive new information, *(e.g., Ticket purchase 30 days before departure);*

Information Component- a cluster of information structure of an Actor or an Object *(e.g., Flight_info, Hotel_adress);*

Information Element- atomic information element that is part an Information Component *(e.g, Flight_number, Nr_of_rooms);*

Action- activity that represents a process component, which is futher decomposable *(e.g., Room_Requisting);*

Elementary Action- activity that represents a process component that is not futher decomposable *(e.g., Cancel_reservation).*

This modeling ideas are necessary for defining domain concept. In SymOntoX are introduced the following binary relations as well: Specialisation, Decomposition, Similarity, Prediction, Relatedness. Their properties are introduced in [17]. The set of concepts, together with their links, allows ontology to be structured according to a semantic network. An example of a concept structured according to OPAL is shown on Figure 2.

Hotel
Def:*A building where trawellers can pay lodging and meals and other services* XML, tag: <*Hotel*> Kind: *Object*
Gen: Accomodation *Spec:* Alpine_hotel, Motel *Similar:* Guest_Farm [0,8] Bed&Breakfest [0,8] *Prediction:* Hotel_Adress,
Hotel_Category *Part of:* Receptivity system *Decomp:* room, restaurant, reception *Relate-objects:* Restaurant *Related/actors:*H_Reservation_Service *Related-processes:* Hotel_Reserving Hotel_Room_Purchasig
(all reported terms, except in the row bellow the concept label, correspond to concepts in the ontology)

Figure 2

An example of a concept Hotel

SymOntoX supports three sorts of users: **User**, with only reading rights; **SuperUser**, who has read and write capabilities; **Ontology Master,** who has the full responsibility on the ontology content, can freely modify it and has the task to validate the concepts proposed by the SuperUsers.

5 Ontology Utilization for Industrial Applications

Mike Ushold and his group has developed in 1996 an EO – Enterprise ontology [18], that is based on Usholds and Kings methodology. The formal EO encoded in OntoLingua is available in the Library of Ontologies that is maintained by Knowledge Systems Lab in Stanford University. Basic concepts of the EO were designed for Enterprise Project requirements, however the EO can be extended and refined for another projects in that domain. The EO defines the following basic terms:

- Activity and Processes – obtains concepts as Activity, Plan, Capability and Resource, that are important for scheduling and planning
- Organization – defines legal entities and structure of the organization
- Strategy – describes purpose, decisions factors and assumptions
- Marketing – is divided into Market and Sales
- Time – includes duration, intervals and time relationship.

Conclusions

The last two hundred years the classical economy recognised only two basic factors of production: labour and capital. Knowledge, productivity, education and intellectual capital were all regarded as exogenous factors. Now technology and knowledge are recognised as key factors of production. The aim of the paper was to underline the importance of the ontology creation in the process of building of knowledge systems that are the step from classical to knowledge economy.

Referencies

[1] Nof S.Y., Morel G., Monostori L., Molina A., Filip F.: From Plant and Logistic Control to Multi-Enterprise Collaboration, In Proc. of the 16th IFAC World Congress, Selected Plenary Milestones and Surveys, July 3-8, 2005, Prague, Czech Republic, pp. 153-166

[2] Schreiber G., Akkermans H., Anjewierden A., de Hoog R., Sgadbolt N., Van de Velde W., Wielinga B., Knowledge Engineering and Management, The CommonKADS Methodology, The MIT Press, 2002

[3] Mizoguchi R.: Part 2: Ontology development, tools and languages. Available at www.ei.sanken.osaka-u.ac.jp/pub/miz/Part2V3.pdf, December 2005

[4] www.iet.com/Projects/RKF/SME/methodologies-for-ontology-development.pdf (Available in December 2005)

[5] http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D1.4-v1.0.pdf, (Available in December 2005)

[6] Fox M. S, Gruninger M.: Enterprise modeling, www.eil.utoronto.ca/enterprise-modelling/papers/fox-aimag98.pdf, (Available in December 2005)

[7] Fernandez-Lopez, M.: Meta-modelling for ontology development and knowledge exchange. In: 15th ECAI Conference Workshop 1 "Ontologies and semantic interoperability" Lyon 2002, p. 6-1

[8] Staab, S. H., Schunurr, R. S., Sure Y.: Knowledge processes and ontologies, IEEE Inteligent Systems, Special Issues on Knowledge Management 16(1), 2001, pp. 26-34

[9] Noy, N. F.: Guidelines to ontology development www.ksl.stanford.edu/people/dlm/papers/**ontology**-tutorial-noy-mcguinness.pdf (Available in December 2005)

[10] http://ontoserver.aifb.uni-karlsruhe.de/ontoedit (Available in December 2005)

[11] http://protege.stanford.edu (Available in December 2005)

[12] http://www.ksl.stanford.edu/software/ontolingua (Available in December 2005)

[13] Missikoff M., Verladi P., Navigli R.: The Usable Ontology: An Environment for Building and Assessing a Domain Ontology, International Semantic Web Conference 2002, Sardinia, Italy

[14] http://www.w3.org (Available in December 2005)

[15] http://www.w3.org/TR/owl-features/ (Available in December 2005)

[16] Bloodsworth P., Greenwood S., COSMOA an ontology-centric multi-agent sstem for co-ordinating medical responses to large-scale disasters, AI Communications 18, 2005, 229-240, IOS Press

[17] Missikoff M., Taglino F., SymOntoX: A Web –Ontology Tools for eBusiness Domains. In the proc. of the Fourth International Conference on Web Information Systems Engineering (WISE'03), p. 343

[18] Ushold M., King. M., Moralee S., Zorgios Y., The Enterprise Ontology, 1995, available at http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html, (Available in December 2005)