Implementation of Knowledge Spaces in Ontologies

Dipl.-Kfm. Yilmaz ALAN
Univ.-Prof. Dr. Stephan ZELEWSKI
Dr. Reinhard SCHÜTTE

Institute of Production and Industrial Information Management
University of Duisburg-Essen - Germany
Universitätsstraße 9
45141 Essen
yilmaz.alan@pim.uni-essen.de
stephan.zelewski@pim.uni-essen.de
reinhard.schuette@pim.uni-essen.de

Abstract

The theory of knowledge spaces by DOIGNON and FALMAGNE is researched on its usability as a formal basis for ontology-based competency profiles. One focus of the research is the formal specification of concepts, which can be used to represent knowledge about the competencies of actors. Another approach is to point out a method to increase efficiency in the acquisition, structuring and representation of knowledge. This shall be achieved by exhausting the potential given by inference rules.

Keywords: knowledge spaces, ontologies, F-logic, competency profiles.

1. Introduction

In order to achieve competitive advantage enterprises increasingly try to integrate activities in their business processes, from which they expect contributions to their *knowledge management*. The background of this approaches is given by the *resource based view* (RBV) [9]. Following the RBV sustainable competitive advantage can only be achieved, when management focuses attention on key resources within the own company. The *competence based view* [3,11] is a special form of the RBV, concentrated on the class of *intangible* assets.

Research in the area of competence based management is as well settled in socio-cultural work as in organizational theories or in IT-solutions for knowledge management. In the framework of the latter, *ontology based* approaches have gained importance in the last years [7,21,23]. Ontologies are used e.g. to structure the concepts which are needed for the description of knowledge about the competencies of the employees ("personal knowledge") or the whole organization ("organizational knowledge"). So the development of ontology-based solutions for competence management systems has been researched with emphasis [15]. From the formal specification of the conceptual structures of competence-management-systems, the foundation of a basis for effective communication among humans and computers is expected.

One of the main advantages which is accompanied with the application of ontologies is the possibility to define *inference rules*, which can be used to add "new" facts to the knowledge base. Although the definition of inference rules is often declared as an important benefit of ontologies in the literature [19], only little attention has been paid to the content of inferential competence management systems. In this article we will try to fill this gap partially by demonstrating an application which is influenced by work in the area of mathematical psychology. The ideas for the study were developed in the framework of a research project[1)] supported by the German Federal Ministry of Education and Research.

The paper is organized as follows: In the next section we will introduce the theory of knowledge spaces. In section 3 we will present ontologies as a method to implement knowledge spaces. In section 4 excerpts from ontologies we have constructed will be shown. Our work closes with an outlook in section 5.

2. Knowledge Spaces

The basis of our study is given by the research of DOIGNON and FALMAGNE [5,17] in the area of *knowledge spaces*. We will propose an interpretation of their work in the field of *competence management*.

According to this, the problems in a domain considered to be relevant by a group of actors constitute the problem set $\mathcal{P}$. The knowledge of an actor empowering him or she to act in order to find solutions for a problem $p \in \mathcal{P}$ - viz his competence - is given by a subset of $\mathcal{P}$. To represent the interdependencies between the problems in the set $\mathcal{P}$, the *surmise relation* $\mathcal{R}$ - given by the subset of the cartesian product of $\mathcal{P}$ with itself - is introduced:

$$\mathcal{R} \subseteq \mathcal{P} x \mathcal{P} \qquad (1)$$

The surmise relation can be interpreted in various ways. In the context of competence management systems the surmise relation will be reasoned by the competence of an actor to solve problems: The relation $(p_1,p_2) \in \mathcal{R}$ will hold, if and only if a solution for a problem $p_2 \in \mathcal{P}$ can be expected from an actor who is capable to find a solution for the problem $p_1 \in \mathcal{P}$.

The characteristics of $\mathcal{R}$ are given as follows:

$$\forall\ p \in \mathcal{P}: (p,p) \in \mathcal{R} \qquad \textit{(reflexivity)}\ (2)$$

$$\forall\ p_1,p_2,p_3 \in \mathcal{P}: (p_1,p_2) \in \mathcal{R} \wedge (p_2,p_3) \in \mathcal{R} \rightarrow (p_1,p_3) \in \mathcal{R} \qquad \textit{(transitivity)}\ (3)$$

1) The official title of our project is KOWIEN (Cooperative Knowledge Management in Engineering Networks). See http://www.kowien.uni-essen.de/ for more details.

$$\forall\, p_1,p_2 \in \mathcal{P}: (p_1,p_2) \in \mathcal{R} \wedge (p_2,p_1) \in \mathcal{R} \rightarrow p_1 = p_2 \qquad (antisymmetry)\ (4)$$

The Relation $\mathcal{R}$ therefore is a *partial order* [4].

Reflexivity includes the assumption that from the capability of an actor to solve a problem p a solution for the problem p can be deduced. This assumption may sound superfluous, but it is necessary for the completeness of the concept. Transitivity on the other hand is of greater importance. It permits the explication of formerly implicit knowledge about indirect relations between problem solving capabilities. *Antisymmetry* is solely needed to preserve the integrity of the concept. It can not be used for the deduction of "new" relations between problems.

The *knowledge state* $\mathcal{KS}$ of an actor is given by the set of problems which he or she is capable to solve. By the introduction of $\mathcal{R}$ some knowledge states get inadmissible. If for example $(p_1,p_2) \in \mathcal{R}$ holds, no actor can have a knowledge state containing p_1 but not p_2.

The set of all admissible knowledge states is defined by

$$\mathcal{KS} \subseteq \mathcal{P} :\Leftrightarrow \forall\, p_1,p_2 \in \mathcal{P}: (p_1 \in \mathcal{KS} \wedge (p_1,p_2) \in \mathcal{R} \rightarrow p_2 \in \mathcal{KS}) \qquad (5)$$

For a domain with $\mathcal{P}$ and $\mathcal{R}$ given as in Figure 1, the *knowledge structure* $\mathcal{K}$ of all admissible knowledge states is defined:

$$\mathcal{K} = \{\emptyset, \{p_2\}, \{p_3\}, \{p_2,p_3\}, \{p_3,p_4\}, \{p_1,p_2,p_3\}, \{p_2,p_3,p_4\}, \{p_1,p_2,p_3,p_4\}\} \qquad (6)$$

While the power set $2^{|\mathcal{P}|}$ of the problem-set $\mathcal{P}$ is 16, the set of admissible knowledge states due to the Surmise-Relation is 8.

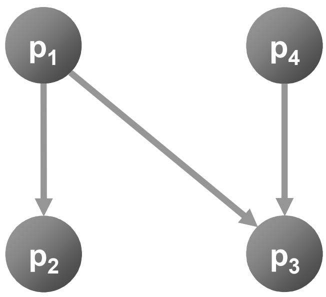

Fig. 1: Extended HASSE diagram for given surmise relation

The set of all admissible knowledge states has two characteristics: Firstly the set is closed under *intersection*:

$$\forall\, \mathcal{KS}_1,\mathcal{KS}_2: ((\mathcal{KS}_1,\mathcal{KS}_2) \in \mathcal{K}) \rightarrow (\mathcal{KS}_1 \cup \mathcal{KS}_2) \in \mathcal{K} \qquad (7)$$

Secondly, the set is closed under *union*:

$$\forall\, \mathcal{KS}_1,\mathcal{KS}_2: ((\mathcal{KS}_1,\mathcal{KS}_2) \in \mathcal{K}) \rightarrow (\mathcal{KS}_1 \cap \mathcal{KS}_2) \in \mathcal{K} \qquad (8)$$

When the latter characteristic is dropped, the concept of *knowledge spaces* can be constructed. It could be argued that the loss of the requirement to be closed under union will weaken the concept. But in the context of competence management systems this will even raise the plausibility of the model: the proof can be given by an example from FALMAGNE ET AL. (1990) [6]: A group of actors with the individual knowledge states $\mathcal{KS}_1, \mathcal{KS}_2, \ldots, \mathcal{KS}_n$ shall therefore be imagined. An actor, who will accumulate the knowledge states of all actors by learning will have the knowledge state $\mathcal{KS}_1 \cup \mathcal{KS}_2 \cup \ldots \cup \mathcal{KS}_n$ after a period of time. A knowledge state being constructed of the intersection of other knowledge states would mean the loss of competencies after the period.

A *Surmise function* σ maps every problem onto a family of subsets of $\mathcal{P}$, which are called *clauses*.

$$\sigma: \mathcal{P} \rightarrow 2^{\mathcal{P}}$$
$$\sigma(p) := \{C_1, C_2, \ldots, C_n\}$$
$$\text{with } C_i \subseteq \mathcal{P}\ (i=1,\ldots,n;\ n \in \mathcal{N}). \qquad (9)$$

It is argued that from the competence to solve a problem $p \in \mathcal{P}$ the competence to solve *all* problems in *at least* one of the clauses from $\{C_1, C_2, \ldots, C_n\}$ can be surmised. The visualization of a knowledge space is given by an AND/OR-Graph [22] as in Figure 2.

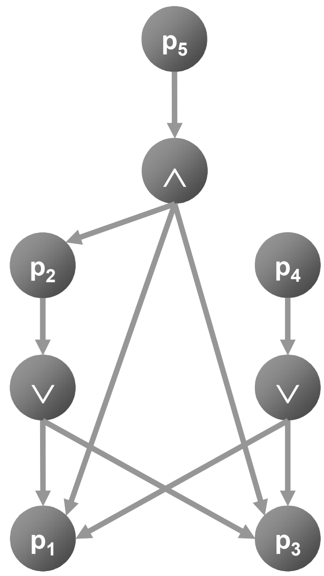

Fig. 2: AND/OR-Graph for visualization of knowledge space

The surmise functions for the knowledge space given in figure 2 are:

$$\sigma(p_1) = \{\{\emptyset\}\}$$
$$\sigma(p_2) = \{\{p_1\},\{p_3\}\}$$
$$\sigma(p_3) = \{\{\emptyset\}\}$$
$$\sigma(p_4) = \{\{p_1\},\{p_3\}\}$$
$$\sigma(p_5) = \{\{p_1,p_2,p_3\}\}$$

In the given example, the value $\sigma(p_2)=\{\{p_1\},\{p_3\}\}$ of p_2 can be interpreted as follows: it is plausible to expect the ability to solve the problems p_1 or p_3 from an actor, who has already proven his competence to solve p_2.

The set $\mathcal{W}$ of all well-formed knowledge states then is given as:

$$\mathcal{W} = \{\emptyset,\{p_1\},\{p_3\},\{p_1,p_2\},\{p_1,p_3\},\{p_1,p_4\},\{p_2,p_3\},\{p_3,p_4\},\{p_1,p_2,p_3\},\{p_1,p_2,p_4\},\{p_1,p_3,p_4\},\{p_2,p_3,p_4\},\{p_1,p_2,p_3,p_4\},\{p_1,p_2,p_3,p_5\},\{p_1,p_2,p_3,p_4,p_5\}\}$$

3. Ontologies

Communication between actors is often affected by ambiguities of natural language. Formal languages have the advantage of providing clearly defined semantics. Ontologies formalize concepts and the relationships between concepts. This formalization can be used for communication purposes.

Research in the area of ontologies is dedicated to the idea of enriching syntactical expressions in a knowledge base with *semantics* [10]. By the formal specification of concepts, knowledge can be structured in ontologies in a machine processible manner. To achieve this goal, the *intensional* content of a concept is explicated on the one hand. On the other hand the *extent* of a concept is given by its instantiation. By the formalization of the intensional as well as the extensional context of concepts an interpersonal agreement on their semantics can be achieved. This shared understanding is needed in every kind of communication.

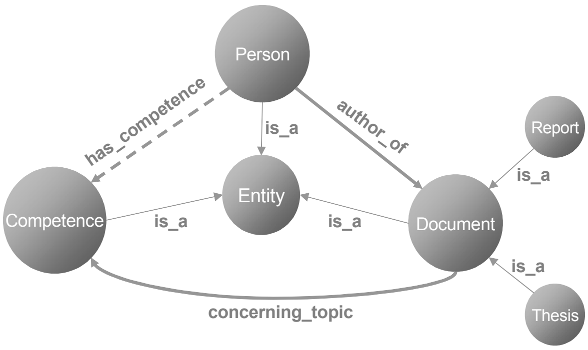

Fig. 3: Informal Ontology

The main components of an ontology are *concepts, relations between concepts* and *axioms* [19]. Figure 3 shows an informal example of an ontology. In the given example, the set of all concepts is given by {*Entity*, *Person*, *Competence*, *Document*, *Report*, *Thesis*}. The set of all relations can be divided into two disjoint subsets. The first subset consists of domain-independent relations. Relations like *is_a* (taxonomic) and *part_of* (meronymy) are examples for this. The second subset consists of domain-dependent relations. In our example, this set is given by {*has_competence*, *author_of*, *concerning_topic*}.

The axiomatic part of an ontology consists of inference rules and integrity rules. By the specification of inference rules formerly implicit knowledge can be explicated. This ability to explicate implicit knowledge is already known from logical programming (e.g. with Prolog). The following example demonstrates the use of inference rules in ontologies:

```
FORALL X,Y,Z
    X[has_competence->>Z] ←
        (X:person[author_of->>Y] AND
        Y: report[concerning_topic->>Z] )
```

The rule fires, when a person (X) is author of a report (Y) and the report concerns a topic (Z). In this case, the fact "person X has the competence Y" will be added to the knowledge base.

Secondly, by the specification of *integrity rules*, inconsistencies in the knowledge-base can be avoided. For example, the rule above can not fire, if the person concerned is explicitly declared to be incompetent for the given problem. Integrity rules are especially required if in a competence management system inference rules are used for plausible deduction. Every kind of plausibility-reasoning lacks generality. In order to maintain *truth* in the knowledge base, integrity rules can be used to avoid false deductions.

The specification of ontologies is done using a formal language. In our approach we have used F-logic [13], since the ability to specify inference rules is mandatory for the implementation of knowledge spaces. Alternative languages like DAML+OIL [12] lack the ability to define domain-dependent inference-rules. In order to maintain interoperability with other works, structural parts of the ontologies have also been embedded in RDF(S) [2,16].

4. Implementation of knowledge spaces in Ontologies

In this section we will present excerpts from ontologies, we have constructed[2)]. The construction of the ontologies is mainly realized using the ontology engineering environment OntoEdit [20,26]. We chose OntoEdit for the specification because of its appropriateness in the different phases of ontology construction and its outstanding inferencing capabilitiy.

In the first part we will introduce the specification of surmise relations. In the second part we will show a way to specify surmise systems.

Implementation of the Surmise-Relation

For an ontology based representation of a domain, the concepts and the relationships between the concepts have to be specified. The F-Logic specification of $(p_1,p_2) \in \mathcal{R}$ is:

P1:competence[has_surmise_conclusion->>P2] (10)

2) See www.pim.uni-essen.de/mitarbeiter/pimyial/Kompetenzontologie.flo or http://www.pim.uni-essen.de/mitarbeiter/pimyial/Kompetenzontologie.rdf for the full ontologies

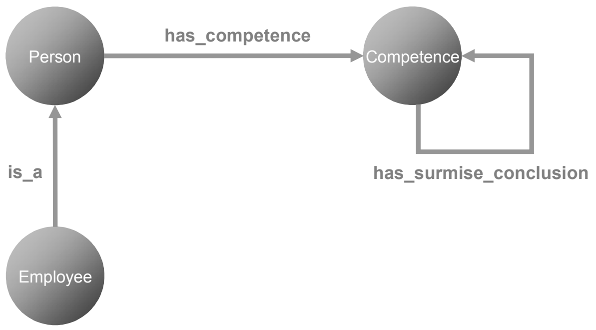

Fig. 4: Structure of the surmise relation ontology

The characteristics of the surmise relation can be implemented as follows: If $(p_1,p_2) \in \mathcal{R}$ is given, it can be formulated:

P1[has_surmise_conclusion->>P2] (11)

The *reflexivity* of the surmise relation is specified as:

FORALL P
P[has_surmise_conclusion->>P] ←
P:competence. (12)

The *transitivity* is specified as:

FORALL P1, P2, P3
P1[has_surmise_conclusion->>P3] ←
P1:competence[has_surmise_conclusion->>P2]
AND
P2:competence[has_surmise_conclusion->>P3]. (13)

While reflexivity and transitivity can be specified by mapping the logical formulas onto F-Logic, the specification of antisymmetry is problematic: the inherent nature of antisymmetry does not include the deduction of "new" facts. Rather it postulates the preclusion of "forbidden" facts. The preclusion of facts exceeds the expressiveness of systems based on ordinary horn-logic, which is supposed in most logically based knowledge or ontology management systems.

The definition of antisymmetry given in formula (4) seems to have the character of a horn formula. But in fact it is a transformation of the following formula:

$$\forall\ p_1,p_2 \in \mathcal{P}: ((p_1,p_2) \in \mathcal{R} \wedge p_1 \neq p_2)) \rightarrow (p_2,p_1) \notin \mathcal{R}. \quad (14)$$

In the conclusion of formula (14) a negation is given. The expression of negations is not included in horn logics. In order to represent the effect of antisymmetry we introduced a relation "*has_not_surmise_conclusion*" being complementary to the surmise relation "*has_surmise_conclusion*". Antisymmetry can then be expressed as follows:

FORALL P1,P2
P2[has_not_surmise_conclusion->>P1] ←
(P1:competence[has_surmise_conclusion->>P2]
AND
NOT equal(P1,P2)).

A different approach would have been a recourse to *propositional logics*. In this case a statement "*inconsistent*" is defined, which fires, when the preconditions of antisymmetry are harmed:

FORALL P1,P2
inconsistent ←
P1[has_surmise_conclusion->>P2]
AND
P2[has_surmise_conclusion->>P1]
AND
NOT equal(P1,P2).

To handle the specification of partial order relationships, we recommend their *reification*. When reifying a relationship the borders of first order logic are exceeded. In first order logic only the quantification over objects is allowed. Instead in the following formula for transitivity we have a quantification over terms (*X,Y,Z*) and over a relation (*REL*):

FORALL X,Y,Z,REL
X[REL->>Z] ←
X[REL->>Y]
AND
Y[REL->>Z]
AND
transitive(REL).

When having specified the formal quality of transitivity, we can instantiate the meta relation "*transitive*" with the object relation "*has_surmise_conclusion*":

transitive(has_surmise_conclusion)

This approach has various advantages: The reuse of parts of the ontology is supported, since the qualities of relationships are defined in a domain-independent way. Furthermore the compactness and clarity of the ontology is raised.

To connect the specification of the surmise relation with concepts concerning actors, we have the following formula:

FORALL A,P1,P2
A[has_competence->>P2] ←
A:actor[has_competence->>P1]
AND
P1:competence[has_surmise_conclusion->>P2].

The concepts of the ontology can then be instantiated in the knowledge base. To query the ontology based model, we again formulate an inference rule. But in this case, the rule has no conclusion. If e.g. the competency profile of an actor named "John" is asked, the following query can be used:

FORALL X
John:actor[has_competence->>X].

Implementation of the Surmise-Function

As stated above, the surmise relation indicates compulsory connections between problem solving competencies. Instead the surmise function includes *optional* relations.

For the implementation of surmise functions, we have specified the concept "*Clause_Set*". For every competence, the corresponding clause set is given. The elements of every clause-set are clauses. Every clause contains competencies.

The concepts and relations of the surmise-system ontology are visualized in Fig. 5.

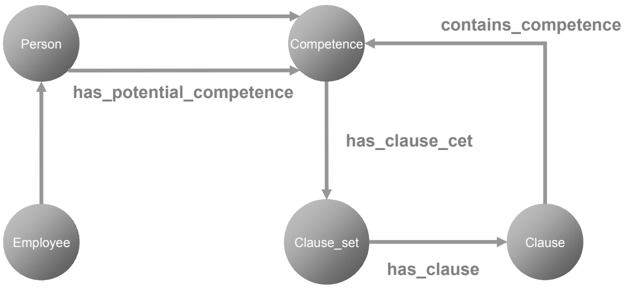

Fig. 5: structure of the surmise-system-ontology

In the surmise system ontology, the deduction of "new" competencies of an actor is not as sound as in the surmise relation ontology. Due to the limitations of horn-logics [18], we formalized the system as follows:

FORALL A1, P1, P2,
Clause_Set1,Clause1
A1[has_potential_competence->>P2] ←
(A1:actor
[has_competence->>P1]
AND
P1:competence
[has_clause_set->>Clause_Set1]
AND
Clause1:clause
[contains_competence->>P2])

The epistemic quality of the formula above is lower then the epistemic quality of the formula specified for the surmise relation ontology, since only a hint for *potential* competencies is given. The reason for this is the known trade-off between *traceability* and *expressiveness* of knowledge representation languages. The surmise system ontology allows the expression of relations between concepts beyond the surmise relation. On the other hand, the facts deducible by the axiomatic structure of the surmise function have a lower quality.

5. Outlook

Ontologies can be used to structure knowledge about the competencies of actors. Besides the effect of avoiding inefficiencies caused by different languages the axiomatic part of ontologies raises model quality. By the implementation of knowledge spaces in ontologies an efficient way is provided to deduce facts concerning the competencies of actors.

Beyond the implementation of the theory of knowledge spaces for competence management systems, several other scenarios are imaginable. These include the following examples:

- *technical fault diagnosis systems*
  When potential sources of errors are specified in a partial order, fault diagnosis can be executed more efficiently. System analysts would be given a possibility to define explicit facts in order to minimize the space for potential errors.
- *medical diagnosis systems*
  Similar to the case of technical fault diagnosis systems, medical diagnosis systems can be improved by knowledge spaces. Ontology based expert systems can aid medics in diagnosing patients. Implicit knowledge about potential causes of diseases can be deduced by specifying relations between symptoms similar to knowledge spaces.
- user modeling for *e-learning systems*
  Traditional e-learning-systems only support sequential learning phases. When users are profiled using an order similar to knowledge spaces, redundant phases could be missed out [1,24].

Furthermore a connection with *Formal Concept Analysis* (FCA) [8] is possible. Following this approach, the intension as well as the extension of a concept is explicitly specified in a *knowledge context*. Methods developed in the framework of FCA, can then be effectively applied to the theory of knowledge spaces [25].

A main critique on the theory of knowledge spaces is the lacking of empirically validated examples. While theory has developed a solid ground for efficient knowledge acquisition, in practice the approach has been neglected in the last years. Studies made under the topic of knowledge spaces are concerned with well-structured expertise areas, like elementary geometry [14]. In order to advance applicability of the theory, it has to be adjusted to ill-structured domains, too. The key success factors for the theory will be the availability of reusable knowledge spaces.

6. References

[1] Albert, D.: Schrepp, M.: Structure and Design of an Intelligent Tutorial System Based on Skill Assignments. In: Albert, D.; Lukas, J. (Eds.): Knowledge Spaces: Theories, Empirical Research, and Applications. Mahwah 1999, pp. 179-196.

[2] Brickley, D.; Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft 30 April 2002. URL: http://www.w3.org/TR/rdf-schema/

[3] Coates, T.T.; McDermott, C.M.: An exploratory analysis of new competencies: a resource based view per-

spective. In: Journal of Operations Management, Vol. 20 (2002), pp. 435-450.

[4] Darnel, M.R.: Theory of lattice ordered groups. New York et al. 1995.

[5] Doignon J.P.; Falmagne, J.C.: Knowledge Spaces. Berlin et al. 1999.

[6] Falmagne, J.C.; Doignon, J.P.; Koppen, M.; Villano, M.; Johannesen, L.: Introduction to Knowledge Spaces: How to Build, Test, and Search Them. In: Psychological Review, Vol. 97 (1990), No. 2, pp. 201-224.

[7] Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Berlin et al. 2001.

[8] Ganter, B.; Wille, R.: Formal Concept Analysis. Berlin et al. 1999.

[9] Grant, R.M.: The Resource-Based Theory of Competitive Advantage: Implications for Strategy Formulation. In: California Management Review, Vol. 33 (1991), No. 3, pp. 114-135.

[10] Guarino, N.: Formal Ontology and Information Systems. In: Guarino, N. (Eds.): Formal Ontology in Information Systems. Proceedings of the First International Conference (FOIS ´98), June 6-8, Trento, Italy. Amsterdam et al. 1998, pp. 3-15.

[11] Herling, R.W.; Provo, J.: Knowledge, Competence, and Expertise in Organizations. In: Herling, R.; Provo, J. (Eds.): Strategic Perspectives on Knowledge, Competence, and Expertise. Baton Rouge 2000. pp. 1- 7.

[12] Horrocks, I.: DAML+OIL: A Reason-able Web Ontology Language. In: Jensen, C.S.; Jeffery, K.G.; Pokorný, J.; Saltenis, S.; Bertino, E.; Böhm, K.; Jarke, M. (Eds.): Advances in Database Technology - EDBT 2002, 8th International Conference on Extending Database Technology, Prague, Czech Republic, March 25-27, Proceedings. Berlin et al. 2002, pp. 2-13.

[13] Kifer, M; Lausen, G.; Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages. In: Journal of the ACM, Vol. 42 (1995), No. 4, pp. 741-843.

[14] Korossy, K.: Modeling Knowledge as Competence and Performance. In: Albert, D.; Lukas, J. (Eds.): Knowledge Spaces: Theories, Empirical Research, and Applications. Mahwah 1999, pp. 103-132.

[15] Lau, T.; Sure, Y.: Introducing Ontology-based Skills Management at a large Insurance Company. In: Glinz, M.; Müller-Luschnat, G. (Eds.): Modellierung 2002. Arbeitstagung der Gesellschaft für Informatik e.V. (GI). Bonn 2002, pp. 123-134.

[16] Lasilla, O.; Swick, R.R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999.
URL: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[17] Lukas, J.; Albert, D.: Knowledge Structures: What They Are and How They Can be Used in Cognitive Psychology, Test Theory, and the Design of Learning Environments. In: Albert, D.; Lukas, J. (Ed.): Knowledge Spaces: Theories, Empirical Research, and Applications. Mahwah 1999, pp. 3-12.

[18] Ma, S.; Sui, Y.; Xu, K.: The Limits of Horn Logic Programs. In: Stuckey, P.J. (Eds.): Proceedings of the 18th International Conference on Logic Programming (ICLP), Denmark. Berlin et al. 2002, pp 467.

[19] Maedche, A.: Ontology Learning for the Semantic Web. Boston et al. 2002.

[20] Maedche, A.; Schnurr, H.P.; Staab, S.; Studer, R.: Representation Language-Neutral Modeling of Ontologies. In: Ebert, J.; Frank, U. (Eds.): Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik. Beiträge des Workshops „Modellierung 2000“, St. Goar, 5.-7. April 2000, Koblenz 2000. pp. 129-142.

[21] Mizoguchi, R.; Kitamura, Y.: Foundation of Knowledge Systematization: Role of Ontological Engineering. In: Rajkumar, R. (Ed.): Industrial Knowledge Management: A Micro-Level Approach, Berlin et al. 2001, pp. 17-36.

[22] Nilsson, N.J.: Principles of Artificial Intelligence. Berlin et al. 1982.

[23] O´Leary, D.: Using AI in Knowledge Management: Knowledge Bases and Ontologies. In: IEEE Intelligent Systems, Vol. 13 (1998), No. 3, pp. 34-39.

[24] Peylo, C; Thelen, T.: Skills und Konzepte als Grundlage für Benutzermodellierung in einem Prolog ITS. Technical Report of the Institute for Semantic Information Processing, Osnabrück 2000.

[25] Rusche, A.; Wille, R.: Knowledge Spaces and Formal Concept Analysis. In: Bock, H.H.; Polasek, E. (Eds.): Data Analysis and Information Systems. Berlin et al. 1996, pp. 427-436.

[26] Sure, Y.; Erdmann, M.; Angele, J.; Staab, S.; Studer R.; Wenke, D.: OntoEdit: Collaborative Ontology Engineering for the Semantic Web. In: Horrocks, I.; Hendler, J.A. (Eds.): The Se-mantic Web - ISWC 2002. First International Semantic Web Conference, Sardinia, Italy, June 9-12, 2002. Berlin et al. 2002, pp. 221-235.