Supporting EBAO Planners by Distributed Ontologies and a Template for Objective-Systems

Tobias Lehmann, Universität der Bundeswehr München, 85579 Neubiberg, Bavaria, Germany; E-mail: tobias.lehmann@unibw.edu
Andreas Karcher, Universität der Bundeswehr München, 85579 Neubiberg, Bavaria, Germany; E-mail: andreas.karcher@unibw.edu

ABSTRACT

This article gives an overview on how military planners in multinational environments can be supported by Knowledge Bases based on Semantic Web Technologies. It reflects parts of results generated during a project for the German Center for Transformation of the Federal Armed Forces from 2005 to 2007. Focus of this project was Knowledge Representation and Knowledge Processing to support EBAO-Planners. For this, firstly context-specific problems are introduced, followed by a brief description of latest enabling Technologies for Knowledge-Modelling and Knowledge-Processing. In particular problems of instance-modeling and abstraction as well as some downsides of inference engines based on First-Order-Logic (FOL) compared to Second-Order-Logic are briefly outlined. An SOL based approach for the analysis of objective-systems is conceptually shown finalized by an example from the Effects Based Approach to Operations (EBAO) context. The conclusion identifies upsides as well as problems and subjects to improvement.

1. THE EFFECTS BASED APPROACH TO OPERATIONS

The Effects Based Approach to Operations (EBAO) introduced a new way of strategic planning not only in correlating Actions and Objectives but in holistically considering Actions, Effects and different Objectives [1]. Effects can be here described as the sum of all results which are caused by actions taken within an Area of Interest, the so called *Focus Area*. The aspect of integrating different Agencies involved with different interests, such as Department of Defence, State Department etc., is emphasized within the EBAO[2]. This approach can not be considered really new [3], but last decade's information technologies enabled efficient network communication as well as Knowledge Representation (KR). These laid the cornerstone for a sufficiently effective dealing with the complexity within fields of modern military operations [4]. Amongst others these are standards in network communication, data bases and XML with its derived modelling languages including Ontology-Languages.

Ontologies allows experts to built and maintain domain specific knowledge [5, 6]. Results from NATO's Multi National Experiment 4 (MNE4) which ended in 2005 have shown that centralized and static Knowledgebase- structure as known from classical databases have crucial drawbacks in giving an situation awareness needed by planners within the EBAO [7]. The focus of this paper is to depict the potentials of Ontology technologies in the field of Multi-Agency and Multi-National Knowledge Representation (KR). As a prerequisite for all planning-work, the formalization of objectives and a mechanism to infer dependencies between those (Reasoning) represent the Knowledge Processing part. We focus on the Web Ontology Language (OWL) here, because it is an implementation of Description Logics, a W3C Standard, and well supported by an Open Source Community. In addition to this there are hardly any competitive technologies at the market [39, 40].

2. OWL: A W3C STANDARD FOR KNOWLEDGE REPRESENTATION

Ontology languages such as OWL are usually attributed to Knowledge Processing not so much as to pure Knowledge Representation [8]. OWL provides no executable methods, but requires inference engines [9], rule systems [10] or wrappers to make structure and data accessible for applications [11]. Encapsulating knowledge structure in own language systems leads to more semantically powerful structures. Amongst others property-centric modelling [12] or a dynamic and logic-based inference [13] of class structures introduce new features in Knowledge Representation. Especially the latter feature, which is missing in executable environments such as Object Oriented (OO) Programming Languages [14], can not be transferred to these without loss of expressivity [15]. Besides this flexible mechanism to create structure by descriptions, OWL offers an fully URI-based referencing mechanism for elements on class level and instance level, which leads to an native implementation of distributed – but seemingly local – KBs [16]. This feature is shown in section 2.1. New applications called Alignment and Mapping tools make use of the graph-based [17] structure of OWL Ontologies and apply distance measure methods, developed from the 70'ies on, in order to support (next to others) a semi automated similarity comparison [18] of different structures. A short overview on this topic can be found in 2.2.

2.1 Ontologies: An Distributed and Modular Approach

For decades organisations have been working on Knowledge Base Models to represent the "whole world" within one single Ontology. This central approach leads to inconsistency, redundancy, unintentionally outdated information and other strongly unwanted effects. Technically Gödel proved in 1931 that a formal system suffers from too little expressivity or from contradictious elements [19]. So seemingly the hope for a "world-system" which contains each an every concept in every perspective has maybe not died but at least suffered a severe set-back then. And this took place even long before technology came even close to a level powerful enough to handle an amount of information this large. Individual perspectives or views – in the sense of an understanding of a subject and its properties (as shown in [20]) – can be seen as reason for why a "common sense modelling" can be considered a permanent point of failure [21, 22]. Simply too many partially contradictious views exist, each with different structures, classes and properties.

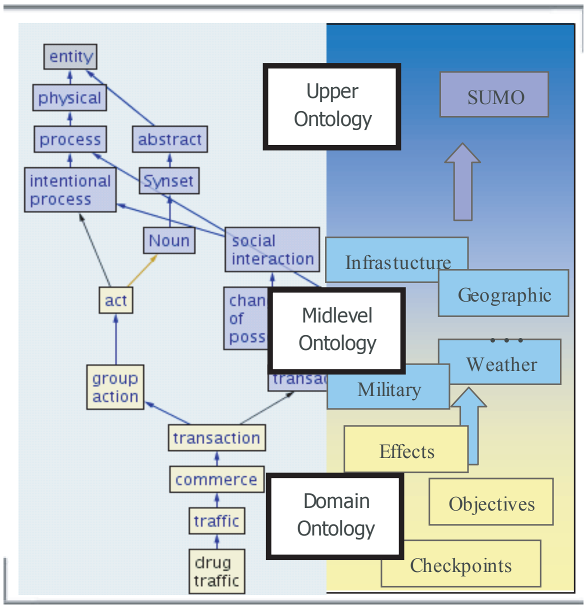
Taking this into consideration, a new way of Knowledge-modelling has been defined. The architecture of ontologies follow a three layer approach, which combines so called upper-level, midlevel and domainlevel ontologies. Once Domain-Knowledge has been made explicit it can be integrated (by means of inheritance) under more general upper Ontologies in order to assemble a KB for special purposes [23]. Figure 1 shows an example Ontology which models Drug Traffic on structure-level. Rectangles depicted in the upper part are more general concepts such as *physical*, *abstract,* or *process* and belong to the upper level of the Standard Upper and Midlevel Ontology (SUMO)-Standard [24].

Rectangles in the centre are of Midlevel, and would not be part of all ontologies, since they contain fairly common but not general concepts. The most specific level, the domain level (lower part), then represents the actual intent of the model [25]. All relations shown in the picture are inheritance relations and therefore enable a switching of different domain level Ontologies since they are of same type by definition (if they inherit from the same midlevel concepts) [26].

2.2 Comparison of Similarity Using Alignment-Tools

In order to participate in a distributed system all elements need to "know" about data structures they are working on [27]. Structures which are of no interest can be neglected but when it comes to data exchange with other participants the meaning

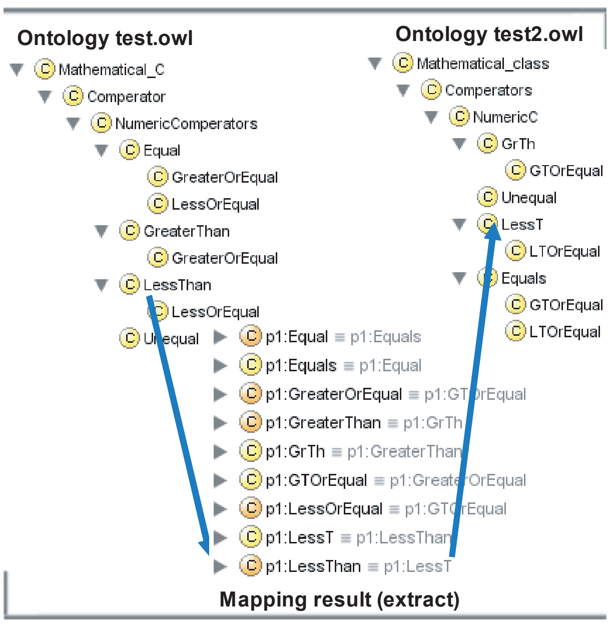
Figure 1. Three-level architecture of an ontology (Source: [25])



of structures and data have to be underpinned by a common understanding. Given the decentralized thought in section 2.1 it becomes highly probable for different actors to work on identical subjects but using different structures. A necessity for bridging these differences appears, if queries are to be used in different environments, class structures are to be aligned or instance-class relations are to be checked (e.g. which class does a given instance belongs to) [28].

Generally speaking differences in models can arise at two conceptual levels: Ether the modelled subject itself (Conceptualisation mismatch), or the way of modelling this subject (Explication mismatch) can differ [28]. While the first challenge is of pure semantic nature, and nearly no support but the recognition of differences can

Figure 2. Example for tool enabled ontology-alignment



be provided, the latter situation is generally supportable by alignment tools [31]. "The way of modelling" in this context means, that a subject can be related to different other subjects on structure level (class definition). Alternatively different modelling-languages can be used. The first case can be solved by a union of all properties, if all concepts' meanings are at least comparable or equal at best [28]. If different languages have been used, a comparison depends on syntax, semantics of primitives, logical representation and the languages' expressivity [32].

Alignment helps integrate the results of different ways of modelling an identical subject under different perspectives using comparable languages. Technically the output of an alignment process is again an ontology which defines relations between classes [29]. In case OWL, specific language-elements such as *owl: sameAs* or *owl:equivalentClass* [12] are used. As an example Figure 2 shows two Ontologies about mathematical comparators and their generated mapping ontology (test.owl and test2.owl).

The result of this process is depicted in the middle and defines equal classes. The limits of alignment are reached, when elements within the compared ontologies are not on semantically comparable levels. In such case some properties would be match one atomic property or class in test2.owl. Integration of this kind can be done syntactically via XSLT processors. Nevertheless this approach seems very promising, since it provides – compared to manual comparison – reasonably support in analysing class dependencies [30].

3. SUPPORTING PLANNERS BY OBJECTIVE-SYSTEMS

All plans try to achieve certain objectives. The definition of objectives, which is depending on time restriction and planning scope, is often done implicitly by planners. This implicit approach seems valid, as long as the planning organization is of limited size. In case of EBAO, several Agencies and Departments are thought to plan and execute as if (at best) they were one single organisation [33]. Strictly formalized information-flows try to limit uncontrolled information exchange among planners of different departments. Hence it is by no means the case that necessary communication between information provider and consumer takes place directly [34]. Needless to say, different departments take a lot of effort to coordinate their planning processes.

But this does not imply persons involved in the planning process to be informed at all times about other-parties' intentions and plans. In order to increase the value of defined and mutually exchanged objectives, one option is to shift these objectives from paper bound media to formalized ontologies. By this, analysis methods can become (partially) IT-enabled – a process highly time consuming, if conducted manually. Of course only a portion of the objective analysis can be done by computers, but areas of high data volumes combined with relatively simple and formalizable relations can be targeted by computers very effectively. One of these areas is the detection of contradictious or depending objectives. The basic question to be answered in this section is in which way a detection of these relations can be achieved. It is not planned to give a detailed introduction into the technical questions of the mechanism itself but more an overview on the idea behind it. The basic assumption for the added value of this mechanisms is, that if planners knew that their actions and effects influence other agencies during planning time they would be able to avoid unwanted (mostly competitive) effects during execution.

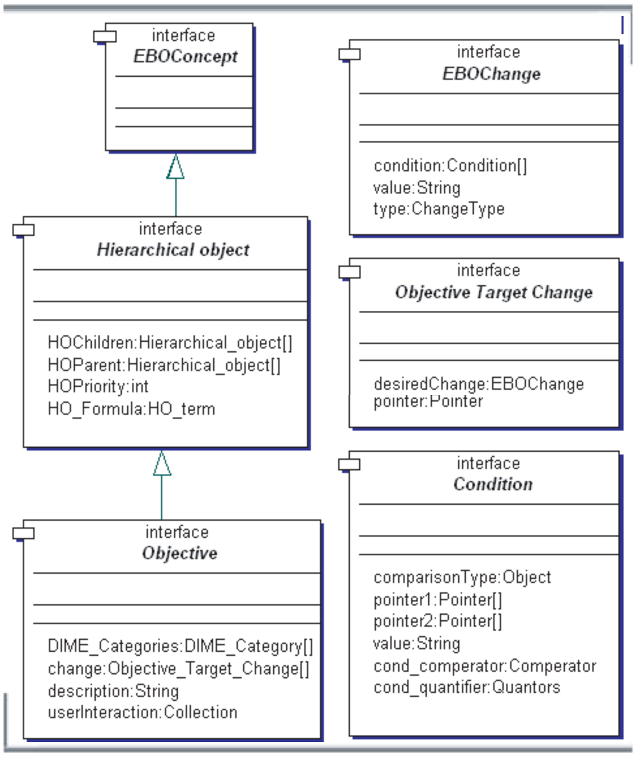
3.1 An OWL- Objective Ontology

A single Objective describes, speaking abstractly in the context of this work, an (internal) state of a set of concepts at a certain point in time. For this an objective needs means to define its target concepts, the desired values to be connected with these and a mechanism to determine if the objective is to be taken into consideration in current planning. The mentioned Ontology counts about 70 classes and 60 properties which are not to be introduced here completely. We focus on core-concepts which are connected to numerous other concepts defining mathematical expressions, actions, user-interaction, or reference constructs.

Figure 3 shows concepts formalized as Unified Modelling Languages (UML) classes.

At the lower left hand side, the Objective-class carries information about EBAO specific categories (*DIME_Category* Diplomatic, Information, Military, Economic), a textual description and Instances for predefined user-interaction. The description of which concept to be changed by which values, is given by references to *ObjectiveTargetChange*-Objects.

Figure 3. Core-concepts of the objective-ontology



ObjectiveTargetChange addresses concepts by *Pointer* Objects (see 3.2). References to *EBOChange*-Objects define desired status and sets of addressed concepts. *EBOChange* itself changes a value in a relative or absolute way (defined by *ChangeType*) to (or by) a certain value. This change is executed, if all conditions are met, while a *Condition*-Object compares two sets of addressed values (see 3.2) or values.

Objectives exit within networked systems of objectives depending on each other [35]. In addition to this it is not a trivial task to define numerically defined end states or to set up references as desired values [36]. Objectives sometimes purely depend on their child-objectives in a sense that if a defined combination of child-objectives is fulfilled then the parent-objective is fulfilled itself. An example for this is given in the following table that defines A as parent-objective for B, C and D (left hand side). The requires-relation on the right hand side determines the conditions under which A is fulfilled (simplified with non-quantified relations only).

childElement(A, B)	requires(A, [B,C])
childElement(A, C)	requires(A, [C])
childElement(A, D)	requires(A, [B,D])

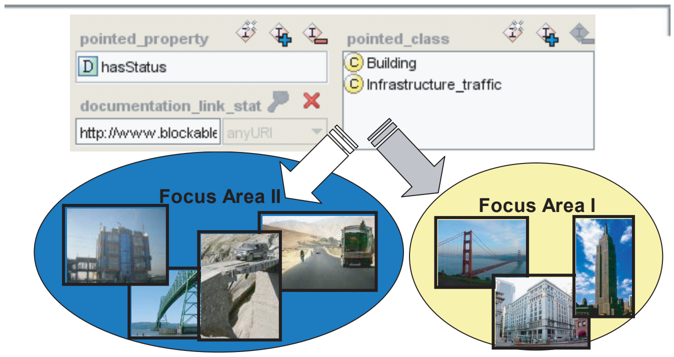
In order to embed single *Objectives* within such systems but keeping the structure encapsulated, the *Objective* class inherits form a class *Hierarchical_object* (see Figure 3). This class provides a special linking mechanism for its subclasses, which defines parent and child-relations as well as priority and *Formulae*. A *Formula* defines the dependencies for all child-elements in a logical expression.

The mechanisms described above are a domain-independent, and therefore exchangeable, way to formalize objectives with flexible addressing mechanism and primitive value or object-comparison mechanism.

3.2 The Advantages of Structural Definitions

One aspect to be outlined more detailed here is the addressing concept used within objective systems. The basic idea behind it is to avoid pure instance reference

Figure 4. Abstract addressing-mechanism: The pointer-class



sources: www.travelblog.org; englishriverwebsite.com; www.arrakeen.ch; bajainsider.com; www.mstl.org; www.ce.ntu.edu.tw; www.corrosioncost.com; www.humanedgetech.com; project1.caryacademy.org

but instead use class references, in order to benefit from the Description Logic (DL) features within OWL. DL allows the definition of so called asserted class-structures enriched by logic statements. These statements can then, after applying reasoning tools, be resolved to new inferred class-structures [13]. Hence a Pointer-instance carries (as shown in Figure 4) a set of classes, one property, a recursive set of pointers and a set of instances to be able to refer to very important instances as well.

In the example shown above all militarily blockable concepts are addressed, which include classes of buildings and traffic- infrastructure. This mechanism is of descriptive nature only and requires an inference engine to determine results of these descriptions. As shown in [37] a class is substituted by all its direct instances and direct instances of its subclasses**.** Exclusive usage of class level concepts supports transferability and reuse of pointers within different Focus Areas.

3.3 Ontologies, Structures and Second Order Logic

If pointers use structure level concepts to address their concepts then First Order Logic does not natively suffice to do the resolution. Technically FOL can query for named predicates. What can not be solved is the question, if any predicate (with variable name) exists which fulfils a goal [38]. Assumed that each instance is transferred to a predicate named after its class and all relations are named after the relations' names and combine domain and range instances, then the predicates' names explicitly carry information about their content.

FOL Representation	SOL option
warlord(X)	concept('warlord', X)
hasInfluence(X, Y)	relation(X, 'hasInfluence', Y)

If OWL-ontologies are transformed as described in [37] all ontology-internal names become variable of a limited set of predicates. This enables SOL-like inferences including reachability analysis and reflection mechanisms based on FOL inference engines.

3.4 Example Results

The following example shows some options how the objective ontology described above can be analysed under conditions of distributed knowledge bases. One of the most important things for a planner setting up objectives is to know, if any other party within his own system has an interest in the concepts addressed by his objectives. If such situation is given, he would also like to know, what kind of dependencies between this two objectives exist. Especially competitive and complementary relations are of interest here, since conducted actions could turn out to be an – seen from the system's point of view – inefficient usage of resources.

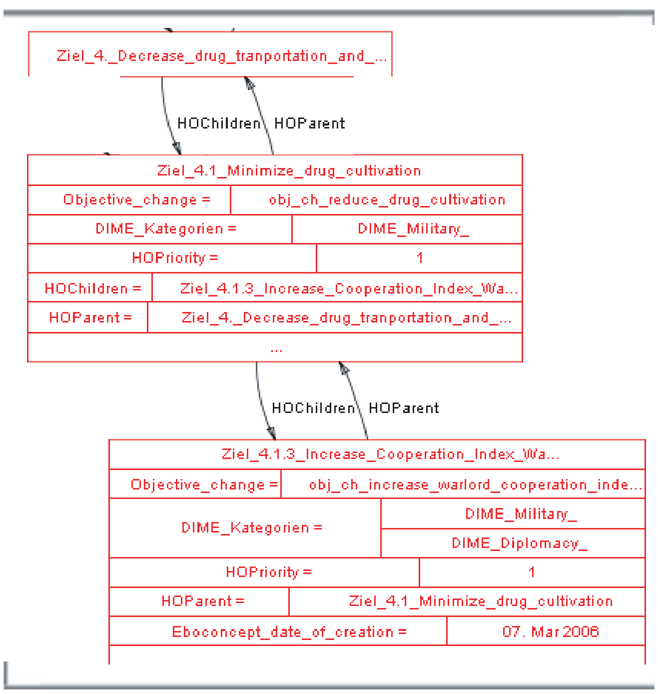
In our example the subject to investigation is drug traffic in the Focus Area of northern Afghanistan. Two System-of-a-System-Analysis analysts [33], one on military, one on diplomatic mission are working on the effects of drug traffic on their afghan scenario. The military planner discovers some warlords' income to be generated in large portions form poppy-cultivation and opium trade. So he creates a rule to make this relation explicit, but due to lack of reliable data she/he does not quantify this information. In addition to this the analyst knows, that some warlords spend quiet a large proportion of their income on running their private militia in order to maintain control over "their" region and to stabilize or exceed their powers. Warlords of great power are considered a threat for the democratically elected afghan government since they tend to subjectively interpret state laws or decisions made by the judicative to their favour.

The diplomatic analyst objective is to increase powerful warlords' will for cooperation with the afghan government. He assumes that without the active participation of these persons, a stable government and prospering economy is far off reach. And she/he knows about the relation between actions taken about the power of warlords and their behaviour connected to the acting parties. Since warlords try to maintain their powers, their willingness to cooperate with anybody trying to weaken their influence decreases. So from the diplomatic perspective (which might be – compared to the military point of view – more far sighted in this case) it seems not recommendable to weaken warlords by aggressive means.

Both analysts now start to develop a formalized objective system to ether (in military case) decrease the income of warlords, or (in diplomatic case) to conserve the current status of the warlords. As a base for their objectives they use the Focus Area Afghanistan 2006 (distributed) KB, which offers a linked system of a large number of (mostly) compatible domain Ontologies. After loading all warlord-relevant parts of this KB to their local computers they discover after some time – indicated by a check for dependencies between their objective-system – that it seems as if somebody else addressed the warlords' income too. Figure 5 depicts a prototype for an objective analyze tool based on Stanford's Protégé-tool. Amongst other features it infers dependencies and contradictions between objectives by comparing addressed sets of concepts and their desired value. Relations such as the dependency between poppy cultivation and a warlord's income can then be inferred. Within a concrete KB these structural descriptions are then projected onto all instances collected by analysts.

Querying details about the discovered references both of them can determine the owner of these relating concepts other objective systems as their source. Now it is apparent, that their objectives are indirectly related to each other, since M wants to prevent poppy cultivation, which is a prerequisite for drug traffic – the basis for warlords' income – and D wants to keep exactly these income at a constant level. After this IT-enabled support, the procedure to resolve this conflict of interests can then be carried out on human to human level.

Figure 5. Determination of contradictious objectives



4. CONCLUSION AND OUTLOOK

The paper presented a conceptual idea on how Ontologies – here implemented in OWL in particular, can be used to mediate KB-models between different agencies. By using the decentralized approach of Ontologies at Upperlevel, Midlevel and Domainlevel dynamic KBs containing detailed expert knowledge can be assembled within a much shorter period of time. Here, alignment tools can provide great help to human analysts. Means of Second Order Logic into play enable a more powerful mechanism for querying and manipulation. Based on SOL domain independent predicates provide a framework for analyzing objective systems on dependencies and contradictions. Although this work has been implemented it has not been tested for scalability and the dynamic loading of instances from different Ontologies. While the first aspect is subject to further research, the latter one will surely be covered by Ontology editors within the near future.

REFERENCES

1. EBO_Prototyping_Team, The Multinational Effects-Based Operations Process. 2005, Washington: Joint Forces Command, Joint Experimentation Directorate.
2. Kemple, W.G., et al. Effects-Based Planning: An Empirical Examination of the Process. in 7th Command and Control Research and Technology Symposium. 2002. Monterey, CA.
3. Clausewitz, C.v., Vom Kriege. 1978, Reinbek: Rowohlt Tb.
4. Curtis, K.P. Knowledge Support for Effect-Based Operations. in CCRTS. 2006. San Diego: CCRP.
5. Smith, M.K., C. Welty, and D.L. McGuinness, OWL Web Ontology Language Guide W3C Recommendation, http://www.w3.org/TR/, 2004, accessed: 2005-09-25
6. McGuinness, D.L. and F.v. Harmelen, OWL Web Ontology Language Overview, http://www.w3.org/TR/owl-features/, 2004, accessed: 2005-11-24
7. Bärtl, C.D.M., et al., MNE 4 KBD Process Limited Objective Experiment - Final Report. 2005, Bundesministerium der Verteidigung: Munich.
8. Koide, S., J. Aasman, and S. Haflich. OWL vs. Object Orientated Programming. in International Semantic Web Conference, Workshop1: Semantic Web Enabled Software Engineering. 2005. Galway: Workshop paper.
9. Friedman-Hill, E., Jess in action. 2003, Greenwich, CT: Manning Publications Co.
10. Horrocks, I., et al., SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/, 2004, accessed: 15.10.2005
11. Knublauch, H., et al., A Semantic Web Primer for Object-Oriented Software Developers, http://www.w3.org/TR/sw-oosd-primer/, 2006, accessed: 2006-09-21
12. Bechhofer, S., et al., OWL Web Ontology Language Reference, http://www.w3.org/TR/owl-ref/, 2004, accessed: 2006-06-30
13. Baader, F., I. Horrocks, and U. Sattler, Description Logics, in Handbook on Ontologies, S. Stab and R. Studer, Editors. 2004, Springer: Berlin Heidelberg New York.
14. Ontoprise_GmbH, How to Write F-Logic Programs, http://www.ontoprise.de/documents/tutorial_flogic.pdf, 2003, accessed: 2006-08-17
15. Jacobs, B., OOP criticism and OOP problems. 2005.
16. Patel-Schneider, P.F., What is OWL (and why should I give a hoot)? http://www-db.research.bell-labs.com/user/pfps/talks/owl/all.html, 2005, accessed: 2005-10-15
17. Diestel, R., Graphentheorie. Vol. 3. 2006, Berlin: Springer.
18. Carroll, J.J., Matching RDF Graphs, http://www.hpl.hp.com/personal/jjc/tmp/matching.pdf, 2001, accessed: 2006-06-21
19. Gödel, K., Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte für Mathematik und Physik, 1931(38): p. 173-198.
20. Brickley, D. and R.V. Guha, RDF Vocabulary Description Language 1.0: RDF Schema, http://www.w3.org/TR/rdf-schema/, 2004, accessed: 2006-06-21
21. Foerster, H.V. and B. Poerksen, Understanding systems: conversations on epistemology and ethics. 2002, New York: Carl-Auer-Systeme.

22. Foerster, H.v., Einführung in den Konstruktivismus. 2005, München: Piper Verlag GmbH.
23. Niles, I. and A. Pease. Towards a Standard Upper Ontology. in 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). 2001. Ogunquit, Maine.
24. Ackermann, J. and e. al., Vereinheitlichte Spezifikation von Fachkomponenten. Gesellschaft für Informatik Arbeitskreis 5.10.3, 2002(Februar 2002).
25. Ševčenko, M., The KSMSA Project, http://virtual.cvut.cz:8080/ksmsaWeb/main, 2004, accessed: 2006-02-07
26. Antoniou, G. and F.v. Harmelen, Web Ontology Language, in Handbook on Ontologies, S. Staab and R. Struder, Editors. 2003, Springer: Berlin Heidelberg.
27. Szyperski, C., Component Software - Beyond Object-Orientated Programming. 2002, New York: Addison-Wesley.
28. Klein, M. Combining and relating ontologies: an analysis of problems and solutions. in IJCAI2001 Workshop on Ontologies and Information Sharing. 2001.
29. Patel-Schneider, P.F., et al., Web Ontology Language (OWL) Abstract Syntax and Semantics, http://www.w3.org/TR/2002/WD-owl-semantics-20021108/semantics-all.html, 2002, accessed: 2006-06-30
30. Bouquet, P., et al., Specification of a common framework for characterizing alignment, knowledgeweb deliverable d2.2.1v2. 2004, Universitat Karlsruhe: Karlsruhe.
31. Bruijn, J.d., et al., D4.2.1 State-of-the-art survey on Ontology Merging and Aligning V1. 2004, Digital Enterprise Research Institute, University of Innsbruck: Innsbruck. p. 99.
32. Euzénat, J., et al. Ontology Alignment with OLA. in 3rd EON Workshop, 3rd Intl. Semantic Web Conference. 2004. Hiroshima (JP).
33. USJFC, JED, and E.P. Team, The Multinational Effect-Based Operations Process, 2005, accessed: 2006-09-21
34. Hill, W., R. Fehlbaum, and P. Ulrich, Organisationslehre 1. Vol. 5. 1994, Stuttgart: UTB.
35. Keeney, R.L., Decisions with multiple objectives. 1978, New York, London, Sydney: John Wiley & Sons.
36. Gregory, G., Decision Analysis. 1988, New York: Plenum Press.
37. Lehmann, T. and A. Karcher. Logic Based Agents and Ontologies Supporting Decision Making in Effects Based Operations. in The 10th World Multi-Conference on Systemics, Cybernetics and Informatics. 2006. Orlando, Florida.
38. Huth, M. and M. Ryan, Logic in Computer Science. 2004, Cambridge: Cambridge University Press.
39. Smith, K.T. and M. Bradley, Lessons Learned in a Department of Defense -The Virtual Knowledge Base. 2006.
40. Corcho, O. and A. Gomez-Perez. A Roadmap to Ontology Specification Languages. in 12th International Conference on Knowledge Acquisition, Modeling and Management (EKAW 2000). 2000. Juan-les-Pins, France: Springer.