Provability-Based Semantic Interoperability via Translation Graphs

Joshua Taylor and Andrew Shilliday and Selmer Bringsjord

`tayloj,shilla,selmer@rpi.edu`
Rensselaer AI & Reasoning (RAIR) Lab
Departments of Cognitive and Computer Science
Rensselaer Polytechnic Institute (RPI), Troy NY 12180, USA

Abstract. *Provability-based semantic interoperability* (PBSI) is a kind of interoperability that transcends mere syntactic translation to allow for robust, meaningful information exchange across systems employing ontologies for which mappings or matchings may not exist, and which can be evaluated by provability-based (PB) queries. We introduce a system of *translation graphs* to formalize the relationships between diverse ontologies and knowledge representation and reasoning systems, and to automatically generate the translation axioms governing PB information exchange and inter-system reasoning. We demonstrate the use of translation graphs on a small number of simple systems to achieve interoperability.

Key words: translation graphs, provability-based semantic interoperability

1 What is Semantic Interoperability?

The proliferation of knowledge-rich systems has led to the creation of myriad intelligent systems possessing diverse reasoning capabilities. Unfortunately, cooperative efforts among these systems are hindered by lack of a common representation scheme or effective general methods for information exchange. Ideally, these systems could reason about their peers' representation schemes and work out a way to exchange information automatically—a capability well beyond the abilities of current systems.

Many systems today achieve various levels of interoperability and information exchange using ontology mapping [1] and schema matching [2]. These techniques are useful and have achieved high levels of information sharing, but cannot capture all the relationships that semantic interoperability requires.

In the general tradition of logicist AI and cognitive science [3,4,5], and specifically in the tradition of logic-based semantic interoperability [6,7], we maintain that semantic interoperability can be evaluated only with respect to provability-based queries. This stems from the fact that ontology mapping and schema matching cannot always capture asymmetry of translation [8], nor can information from a source ontology always be translated into a corresponding form in a target ontology, even if the information has semantic consequences in the target ontology.

Ontologies contain complex relationships among their own terms, and any approach to semantic interoperability must be able to capture not only these, but also the relationships between between multiple ontologies. A system which does not use a sufficiently expressive formalism or language to describe these relationships is inherently specialized and cannot be used for general applications.

Furthermore, consumers of the products of semantic interoperability should have access to the *justifications* that bring about those products. Consumers should have, then, in a schema-mapping approach, access to the mapping itself, in an axiomatic approach, access to the axioms, and in a *provability*-based approach, access to the *proofs*. Ideally, the proofs would be couched in a format that is readily understood by non-specialists; e.g., proofs in natural language are far superior to resolution based proofs. Herein we describe a new brand of PBSI that meets the desiderata just enumerated.

2 The Need For Semantic Interoperability

In a Truly Useful Semantic Web. To achieve a useful Semantic Web, information on the web must (1) be structured in a meaningful way and (2) information from different systems must be able to be combined easily and *meaningfully*. (1) is being addressed as more and more information is stored in databases, and by the adoption of regular markup languages such as XML and XHTML. (2) is only happening partially. Service-oriented architectures *are* sharing information meaningfully, but need complete knowledge of the ontologies employed by the systems involved. When systems with web presence can share information without having to have extensive knowledge of their peers' ontologies or schemata, web-based agents can be built that are capable of deep reasoning and planning [9].

In the Defense and Intelligence Communities. The defense and intelligence communities have, for some time, recognized the need for semantic interoperability, and have sponsored research in tools and languages to address this need. Some results of this research are the DARPA Agent Markup Language [10], DAML, and DAML+OIL, for the markup of information and for the description of ontologies. Though languages such as KIF [11] and Common Logic [12] have been developed for describing the *relationships* between ontologies and for the exchange of information between ontologies, in 2005 the Disruptive Technology Office sponsored the Interoperable Knowledge Representation for Intelligence Support (IKRIS) workshop [13], which resulted in the IKRIS Knowledge Language (IKL), an extension to Common Logic that addresses specific needs of the intelligence community.

3 Relevant Past Approaches

We review a number of past approaches to the problem of semantic interoperability. We note how these approaches fare with respect to the aforementioned desiderata; in particular, whether: an approach is logically based, asymmetry of translation is preserved, information in a foreign ontology can influence a query in a native ontology even when the foreign information cannot be directly translated, and the quality of available justifications is sufficiently high.

When the subject domains and vocabularies of the ontologies to be related are similar and the information represented within them is not too complex, *schema matching* can be effective in translating information from one ontology to another. With schema matching, corresponding terms from the ontologies are selected, and information from one is recast in another. There are automated tools that aid in schema matching [2]. Evaluating whether a schema matching is correct can be difficult, particularly if the matching has been generated (even partially) automatically. A schema matching can be provided as primitive justification for results. It seems difficult, with schema matchings, to capture semantic influence when information translation is not possible.

The use of *schema morphisms* to map the sentences of one ontology to sentences of another allows for more complex transformations between ontologies. This approach can be used when ontologies are treated as *institutions* [14]. Within this framework, it is possible to determine whether a schema morphism is correct [15], and to impose constraints that capture some of the asymmetry of translation and semantic influence. Morphisms are not trivial to construct, but can capture relationships between ontologies using different logics [16]. Signature morphisms are expressed with a different formalism and notation than the ontologies themselves, however, and so the justification for a particular translation requires human intervention or specialized reasoning outside of the ontologies themselves.

Simple syntactic manipulation of sentences does not afford the meaningful translations that are desired. In fact, to answer queries expressed in a query ontology using information from various source ontologies often requires making use of information from many source ontologies. Unfortunately, sentences in a source ontology that have semantic consequences in a target ontology cannot always be translated into the target ontology [8]. Semantic interoperability is still attainable, however, by relating the ontologies logically, and evaluating queries with respect to provability. Ontologies can be related axiomatically using *lifting axioms* [6] or by merging the ontologies to be related and expressing *bridging axioms* in the new merged ontology [7].

The techniques reviewed above have been used in real applications and have successfully enabled varying levels of interoperability. No individual system, however, possesses all of the necessary qualities for top-notch semantic interoperability. Building on these excellent foundations, we believe that our system of translation graphs takes a step closer to the ideal.

4 Formal Preliminaries

The work herein described is, clearly, logic-based, and partakes of the paradigms of logic-based AI [3], cognitive science [4], and computational cognitive modeling [5].

We treat ontologies as pairs of the form $\langle \Sigma, \Phi \rangle$ where Σ is a signature in a many-sorted logic, and Φ is a set of sentences in Σ. While many-sorted logic is not employed by all ontology designers, it is appropriate for describing many ontological constructs including modalities, has a impressive history within computer science and mathematics, and is reducible to standard first-order logic [17].

A *sort* is a domain, a universe, or a set of objects. There is a global set of sorts, S^*. Generally, every signature will contain a sort corresponding to truth values. In traditional logics, this sort is the set $\{\mathbf{true}, \mathbf{false}\}$, but this needn't be the case. Many-valued logics, for instance, will use a different sort for truth values. A *functor* f is a function $s_0 \times \ldots \times s_{n-1} \rightarrow s_n$ where $s_0, \ldots, s_n$ are elements of S^*. $\langle [s_0, \ldots, s_{n-1}], s_n \rangle$ is the *rank* of f and denoted $\mathit{Rank}(f)$.

A *signature* Σ is a tuple $\langle \sigma, \phi \rangle$ where σ is a subset of S^*, called the sorts of Σ and ϕ is a partial injective function from string-rank pairs to functors of the same rank. The range of ϕ is the set of functors of Σ. There is a restriction on ϕ that for every functor f among Σ's functors, each sort in f's rank is one of Σ's sorts.

A well-formed term of Σ has a particular *interpretation* which denotes the application of corresponding functors to their arguments. E.g, if $\mathsf{man}(\mathsf{Socrates})$ is sentence of Σ_1 and $\mathsf{human}(\mathsf{Sokrates})$ is a sentence of Σ_2, but both Σ_1 and Σ_2 map man and human, respectively, to the same functor f of rank $\langle [s_1], s_0 \rangle$, and $\mathsf{Socrates}$ and $\mathsf{Sokrates}$ to the same functor g of rank $\langle [], s_1 \rangle$, then the two sentences have the same interpretation.

5 Ontology Modifications

A number of operations can be defined on signatures which correspond to incremental modifications that might be performed on the signatures of ontologies. Four primitive operations on signatures are defined by the following equations

$$\mathit{AddSort}(s, \langle \sigma, \phi \rangle) = \langle \sigma \cup \{s\}, \phi \rangle \tag{1}$$

$$\mathit{RemoveSort}(s, \langle \sigma, \phi \rangle) = \langle \sigma - \{s\}, \phi \rangle \tag{2}$$

$$\mathit{AddFunctor}(w, f, \langle \sigma, \phi \rangle) = \langle \sigma, \phi \cup \{\langle \langle w, \mathit{Rank}(f) \rangle, f \rangle\} \rangle \tag{3}$$

$$\mathit{RemoveFunctor}(w, r, \langle \sigma, \phi \cup \{\langle \langle w, r \rangle, f \rangle\} \rangle) = \langle \sigma, \phi \rangle \tag{4}$$

subject to several restrictions. $\mathit{RemoveSort}(s, \langle \sigma, \phi \rangle)$ is undefined if any of the functors of $\langle \sigma, \phi \rangle$ use s. $\mathit{AddFunctor}(w, f, \langle \sigma, \phi \rangle)$ is undefined if $\langle w, \mathit{arity}(f) \rangle$ is already mapped to some functor. $\mathit{RemoveFunctor}(w, r, \langle \sigma, \phi \rangle)$ is undefined if ϕ does not map $\langle w, r \rangle$ to any functor.

With the primitive methods, simple ontologies can be constructed that specify only the vocabulary of a language. However, ontology consists not only in vocabulary, but also in the *meaning* of the vocabulary and the relationships among these terms. As a result, many knowledge representation languages include forms analogous to Athena's [18] `define-symbol` for defining symbols *axiomatically.* For instance, $\mathsf{MatGrandmotherOf}(x)$, denoting the maternal grandmother of x can be defined in KIF using $\mathsf{MotherOf}(x)$ by `(deffunction MatGrandmotherOf (x) := (MotherOf (MotherOf x)))`.

Both classical mathematicians and logicians along with modern knowledge representation language designers have devoted a great amount of time to the subject of the forms that can be used in axiomatic definitions. Some definitions may be implemented as macro-like substitutions, while in other cases, the entire axiom must remain available for subsequent reasoning [11, Ch. 11].

6 Translation Graphs

We implemented a prototype of the structures and modifications described in the previous section, thereby providing a framework in which to perform natural ontology-related activities, such as ontology construction and mapping. Ontology construction becomes easy: Starting from an empty signature (i.e., a signature with no sorts or functors), existing ontologies can be recreated by adding the ontology's sorts, and then relations and function symbols. These reconstructed ontologies can then be related by adding the functors of one ontology to another with axiomatic definitions. Displaying the process graphically inspired translation graphs.

After initial experiments demonstrated the feasibility of this approach, we realized that the process could be used to describe the interoperability in the IKRIS workshop and experiments in interoperability between robust software systems, such as Oculus' GeoTime [19,20], SUNY Albany's HITIQA [21], Attempto Controlled English [22,23], and the RAIR lab's own Slate [24] and Solomon [25].

A *translation graph* is a directed graph whose vertices are signatures, and whose edges denote axiomatic relationships between the signatures of the graph. If signatures Σ_i and Σ_j are vertices of some translation graph and the edge $\langle \Sigma_i, \Sigma_j \rangle$ is in the graph, there is information associated with it that describes how information represented in an ontology employing Σ_i can be used in an ontology employing Σ_j. This property is transitive, and so a Σ_u, Σ_v *path* contains information for using information under Σ_u in Σ_v.

7 An Example

We present an example to show that translation graphs can be used to enable interoperability between ontologies whose subject domains intersect but are not identical, that queries can be answered with information from multiple ontologies, and that the information used to answer the query is not representable in all of the ontologies presented. (For the sake of readability and conciseness, we will ignore issues such as namespaces and the use of fundamental datatypes such as strings and numbers.)

We consider four separate software systems operating with four distinct ontologies amongst which information will be shared.

The first two systems are social networking programs which represent information about phone calls. The first system, $\mathcal{A}$, keeps records of the form $\mathsf{Called}(x, y)$ to denote that x called y, where x and y are names of individuals. The second system, $\mathcal{B}$, uses $\mathsf{CalledBy}(x, y)$ to denote that x was called by y, where x and y are names of individuals. $\mathcal{A}$ and $\mathcal{B}$ can be related with the primitive operations described earlier; the result is shown in Figure 1. The function Called is added to $\mathcal{B}$ with an axiomatic definition, yielding an intermediate signature. $\mathsf{CalledBy}$ is removed from the intermediate signature, resulting in $\mathcal{A}$. Tracing the path between the ontologies and collecting axioms along the way gives all the information needed to use information from one ontology in the other.

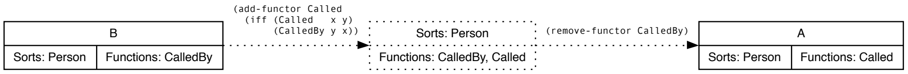

Fig. 1. Ontologies $\mathcal{A}$ and $\mathcal{B}$ are related.

The axiomatic definition between $\mathcal{A}$ and $\mathcal{B}$ is a biconditional and could be optimized as a rewriting rule. That is, assertions in one ontology could be *rewritten* in terms of the other's vocabulary. The translation here is symmetric, and could be handled by schema matching tools.

Next, we introduce a cellular phone company database $\mathcal{C}$ which has information about phone calls made on the cellular network, and keeps records of the form $\mathsf{Phoned}(n_1, n_2)$ where n_1 and n_2 are phone numbers between which calls have been placed. Figure 2 illustrates the relationship between $\mathcal{C}$ and $\mathcal{A}$.

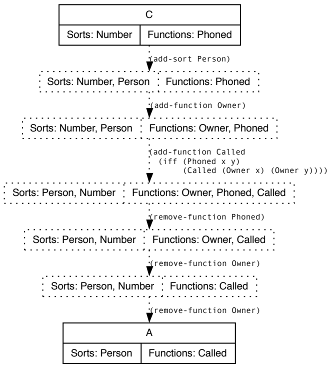

Fig. 2. Phone company $\mathcal{C}$ is related to $\mathcal{A}$.

While no individual link in Figure 2 is particularly complicated, the addition of the axiomatically defined Owner deserves special note. Owner(x) denotes the person who owns a phone number x. Owner is present in neither $\mathcal{A}$ nor $\mathcal{C}$, but its use in relating them does seem clear: Owner functions as a sort of semantic placeholder. Without an interpretation of Owner, information exchange would not be possible; there would be information missing. However, the use of translation graphs has allowed us to capture *what* is needed to exchange information meaningfully.

Another possibility is that Owner may stand for a non-logical function. For instance, in the process of exchanging information, occurrences of Owner(x) might be replaced with the results of a database lookup or some procedural transformation (e.g., if phone numbers were a function of the characters comprising a person's name).

In this example, however, we integrate the database of a reverse phone number lookup system, $\mathcal{D}$. In this case, the information that $\mathcal{D}$ provides is not phone records, but pairs of phone numbers and their owners' names. $\mathcal{D}$ records that Owns(x, y) when x, a person, owns the phone number y. The integration, shown in Figure 3, is straightforward.

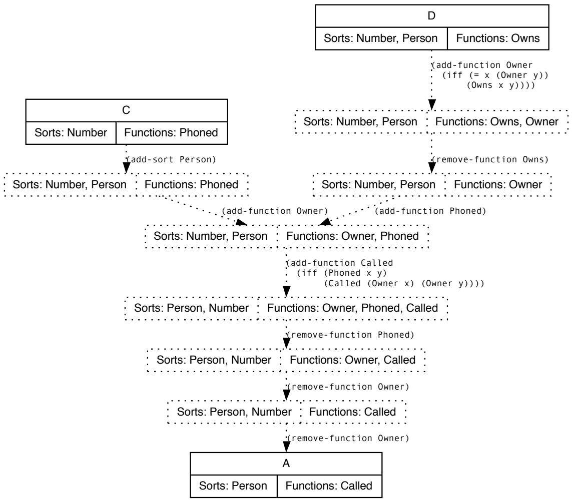

Fig. 3. The information in $\mathcal{D}$ is made available to $\mathcal{A}$ and $\mathcal{C}$.

Having connected $\mathcal{A}$ with $\mathcal{B}$, and then $\mathcal{A}$, $\mathcal{C}$, and $\mathcal{D}$, enough work has been done to yield the translation graph shown in Figure 4. The graph can be used to describe the relationships between the ontologies, and the axiomatic relationships needed to answer queries about the contents of the four knowledge bases can be automatically extracted from it.

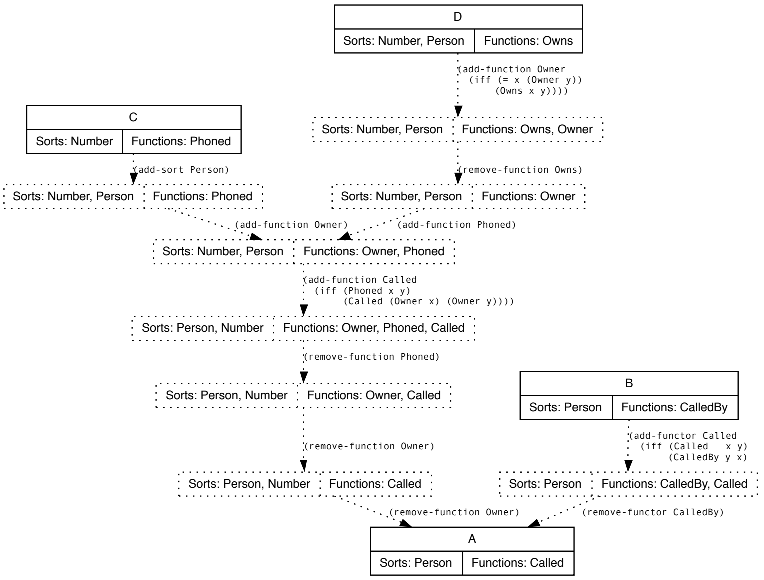

Fig. 4. The final translation graph of the relationships between the systems.

Remarks. In such a small example, the overall *structure* of the translation was not given much thought. In real systems, however, engineers must consider the implications of their translation structures. For example, in some situations, an interlingua and intertheory may be preferred, or in some cases it may not be appropriate or feasible [26,27]. However, we present translation graphs without expressing preference among these possible architectures; translation graphs general enough to be applied in an architecture-agnostic manner.

With the translation graph as given, it would be possible to run automated reasoners directly on the union of the knowledge bases and all the axioms extracted from the edges of the graph. Of course, intractability and undecidability make this a tricky technique, but there is an interesting parallel to Green's method. Green's method extracts plans that achieve particular goals from proofs that such plans exist [28]; with the naive method above, interoperability and translation are achieved as a *side effect* of automated theorem proving.

8 Conclusion and Future Work

Automaticity. The ultimate dream of this sort of R&D is full automaticity. Following a divide and conquer approach, translation graphs allow for the automatic production of bridging axioms. So, if translation graphs could be automatically produced, the dream would be reality. We are investigating the application of automatic programming [29] toward this goal. More immediately, some of the approaches in automated schema matching could be applied.

Sophisticated Ontology Representation. We built translation graphs with the signatures of many-sorted logic as nodes, for flexibility and convenience of expression, though such graphs lack some desirable features such as subsorting, sort hierarchies, and a standard language for describing the signatures themselves. There has been a great deal of research in what kind of reasoning [30] must be performed over ontologies [31], and there are many languages, such as RDF, DAML, and OWL, designed for the purpose of ontology description. Building translation graphs from ontologies represented in these languages would allow us to work with many ontologies already constructed and in use today.

Categorizing Axiomatic Definitions. From certain types of axiomatic definitions we can extract rewriting rules (inline translations); indeed, to make the translation graph approach scale well, optimizations such as inline translations are almost certainly necessary. We believe more sophisticated rewriting rules and other types of procedures can be developed by examining *paths* in a translation graph, and will be pursuing this line of work.

Acknowledgements. With much gratitude, we acknowledge the financial support provided by the Disruptive Technology Office (DTO), contract # N61339-06-C-0128 to RPI, A-SpaceX program.

References

1. Choi, N., Song, I.Y., Han, H.: A survey on ontology mapping. SIGMOD Rec. **35**(3) (2006) 34–41
2. Wang, G., Goguen, J.A., Nam, Y.K., Lin, K.: Critical points for interactive schema matching. In Yu, J.X., Lin, X., Lu, H., Zhang, Y., eds.: Advanced Technologies and Applications: Proceedings of Sitch Asia Pacific Web Conference, Hangzhou, China, Springer (2004) 654–664
3. Bringsjord, S., Ferrucci, D.: Logic and artificial intelligence: Divorced, still married, separated...? Minds and Machines **8** (1998) 273–308
4. Bringsjord, S., Yang, Y.: Representations using formal logics. In Nadel, L., ed.: Encyclopedia of Cognitive Science Vol 3. Nature Publishing Group, London, UK (2003) 940–950
5. Bringsjord, S.: Declarative/logic-based computational cognitive modeling. In Sun, R., ed.: The Handbook of Computational Cognitive Modeling. Cambridge University Press, Cambridge, UK (forthcoming)
6. Buvač, S., Fikes, R.: A declarative formalization of knowledge translation. In: CIKM '95: Proceedings of the fourth international conference on Information and knowledge management, New York, NY, USA, ACM Press (1995) 340–347
7. Dou, D., McDermott, D., Qi, P. Whitestein Series in Software Agent Technologies and Autonomic Computing. In: Ontology Translation by Ontology Merging and Automated Reasoning. Birkhäuser Basel (2005) 73–94
8. Dou, D., McDermott, D.: Deriving axioms across ontologies. In: AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM Press (2006) 952–954
9. Hendler, J.: Agents and the Semantic Web. IEEE Intelligent Systems (2001) 30–37

10. DARPA Agent Markup Program: DARPA Agent Markup Language Homepage (2003) http://www.daml.org/.
11. Genesereth, M.R., Fikes, R.E.: Knowledge Interchange Format Version 3 Reference Manual. (1997)
12. Common Logic Working Group: Common Logic Standard. http://cl.tamu.edu/ (2007)
13. MITRE: IKRIS, workshop site: http://nrrc.mitre.org/NRRC/ikris.htm (2007)
14. Goguen, J.A., Burstall, R.M.: Introducing institutions. In Clarke, E.M., Kozen, D., eds.: Logics of Programs. Volume 164 of Lecture Notes in Computer Science., Springer (1984) 221–256
15. Goguen, J.A.: Information integration in institutions. Paper for Jon Barwise memorial volume edited by Larry Moss (2004)
16. Goguen, J.A.: Data, Schema, Ontology and Logic Integration. Logic Journal IGPL **13**(6) (2005) 685–715
17. Manzano, M.: Extensions of First Order Logic. Cambridge University Press (1996)
18. Arkoudas, K.: Athena. http://www.cag.csail.mit.edu/˜kostas/dpls/athena (2005)
19. Kapler, T., Harper, R., Wright, W.: Correlating Events with Tracked Movements in Time and Space: A GeoTime Case Study. In: Proceedings of the 2005 International Conference on Intelligence Analysis (IA 2005), McLean, VA, USA (2005)
20. Chappell, A., Bringsjord, S., Shilliday, A., Taylor, J., Wright, W.: Integration Experiment with GeoTime, Slate, and VIKRS. ARIVA Principle Investigator Meeting Handout (2007)
21. Strzalkowski, T., Small, S., Hardy, H., Yamrom, B., Liu, T., Kantor, P., Ng, K.B., Wacholder, N.: HITIQA: A Question Answering Analytical Tool. In: Conference Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, VA, USA, MITRE, Sponsored by the Office of the Assistant Director of Central Intelligence for Analysis and Production (2005)
22. Fuchs, N.E., Kaljurand, K.: Attempto Controlled English: Language, tools and applications. Lecture / Presentation (2006)
23. Bringsjord, S., Arkoudas, K., Clark, M., Shilliday, A., Taylor, J., Schimanski, B., Yang, Y.: Reporting on some logic-based machine reading research. In: Proceedings of the 2007 AAAI Spring Symposium on Machine Reading. (2007)
24. Bringsjord, S., Shilliday, A., Taylor, J.: Slate, http://www.cogsci.rpi.edu/slate/ (2007)
25. Bringsjord, S., Clark, M., Shilliday, A., Taylor, J.: Solomon, http://www.cogsci.rpi.edu/solomon/ (2007)
26. Barker-Plummer, D., Greaves, M.: Architectures for Heterogeneous Reasoning on Interlinguae (1998)
27. Menzel, C.: Common Logic Standard. In: Santa Fe 2003 Metatdata Forum Symposium on Ontologies. (2003) Presentation of the case for Common Logic.
28. Genesereth, M.R., Nilsson, N.J.: Logical foundations of artificial intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1987)
29. Rich, C., Waters, R.C.: Automatic programming: Myths and prospects. Computer **21**(8) (1988) 40–51
30. Antoniou, G., Franconi, E., van Harmelen, F.: Introduction to semantic web ontology languages. In Eisinger, N., Małuszyński, J., eds.: Reasoning Web, Proceedings of the Summer School, Malta, 2005. Number 3564 in Lecture Notes in Computer Science, Berlin, Heidelberg, New York, Tokyo, Springer-Verlag (2005)
31. Smith, B.: The basic tools of formal ontology. In Guarino, N., ed.: Formal Ontology in Information Systems, IOS Press (1998) 19–28