Answer Set Programming for Intelligent Web Services

Thomas Eiter, Michael Fink,
Roman Schindlauer, and Hans Tompits

Technische Universität Wien,
Institut für Informationssysteme,
Abt. Wissensbasierte Systeme 184/3,
Favoritenstrasse 9-11, A-1040 Vienna, Austria
email: [eiter,michael,roman,tompits]@kr.tuwien.ac.at
URL: www.kr.tuwien.ac.at

Abstract

As the Web evolves from a content aggregation designed for human consumption to a machine-understandable data collection, a Semantic Web will come into existence, serving as a infrastructure for intelligent Web services. Besides the syntactic and semantic groundwork of data representation, powerful and versatile reasoning techniques are in demand to enable next generation intelligent agents and Web services. To solve specific problems in this domain, enhanced techniques are needed facilitating declarative formalisms and the processing of incomplete or inconsistent information. Answer set programming (ASP) is a new, promising paradigm in which these issues can be addressed. In this paper, we present two projects which take advantage of advanced reasoning capabilities to implement powerful information services.

1 Introduction

The significant boom of the World Wide Web made a vast amount of information available to a large group of users. However, most of this information is neither structured nor centrally accessible, thus creating the need for suitable tools and methods for building an information processing infrastructure. One approach is the use of multi-agent systems, where special *information agents* [4] provide various services to find, select, and query relevant information sources. The agents are considered to be autonomous entities, such as software programs or software robots. The user is presented a uniform interface for accessing all available services and information sources, without having to bother with the heterogeneity underneath. The system takes care of choosing the appropriate sources, accessing them, and returning the desired information.

The realization of such systems requests for special functionalities and capabilities, which emerge from specialized tasks like search and assessment of information sources, query planning, and information merging and fusion, thereby dealing with incomplete information and inconsistencies [18, 10]. Such capabilities are provided as services on request by various kinds of information agents, which usually form a society of agents for cooperatively solving complex information-retrieval problems. It is not hard to imagine that an advanced approach to any of these problems must involve, in some form, logical reasoning tasks, based on ad hoc knowledge about the task in question and on background knowledge of the domain, suitably represented in a knowledge base. The goal of our research is to investigate how methods from declarative logic programming can be employed to allow information agents to solve some of their tasks more "intelligently".

Some of the tasks for intelligent information agents have already been widely addressed and solutions have been developed. Existing intelligent Web systems have their main focus in some of the following fields:

User modeling and profiling: addressing the issue of deploying adaptive user interfaces and recommendation systems, integrating user queries on the basis of a user profile or directly suggesting the user items of interest.

Analysis and preprocessing of information sources: building up, on the basis of an application domain description, the meta-knowledge on which reasoning and decision making is based.

Information integration and information management: covering a wide area of applications and different problems.

Especially in the field of information integration, a large number of systems and frameworks has been developed (for an overview cf. [10, 16]).

2 Intelligent Web and Information Services

The major drive of Semantic Web development are its potential benefits in offering (more) intelligent Web services. Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. In the Web services model, the user is not browsing hypertext information any more, but calling remote procedures, viewing the Web as a distributed device of computation rather than a collection of information. So far, the focus is mainly on providing an XML based infrastructure by technologies like UDDI (Universal Description, Discovery, and Integration) [23], WSDL (Web Service Definition Language) [24], and SOAP (Simple Object Access Protocol) [22]. These standards address message definition and exchange tasks; additional support is needed for more sophisticated processes like automatic Web service discovery, selection, mediation, and composition into complex services. The approach taken by the SWWS (Semantic Web enabled Web Services) project [13] is to combine Web services with semantic Web technologies to provide mechanization in service identification, configuration, comparison, and combination.

In order to provide Web services, collaborating software agents have been applied very successfully (for an extensive overview of agent programming approaches, cf. [19]). Several models of multi-agent systems have been developed during the last years, together with a number of successful applications, mainly in the fields of heterogeneous information and software integration, electronic commerce, and Web interfaces. A common practice of service provision in a multi-agent setting is a yellow pages service, where individual agents register their capabilities in order to offer a particular service (see e.g., the *IMPACT* agent platform [21]).

One essential feature of Web services is information integration—the task of providing a uniform interface to various pre-existing data sources, so as to enable users to focus on specifying what they want, rather than thinking of how to obtain the answer. Thus, an information integration system frees the users from the tasks of finding the relevant data sources, interacting with each source in isolation, and selecting, cleaning, and combining data from multiple sources.

Our activities focus on *knowledge-based approaches* for developing reasoning components for intelligent information access. In particular, we are interested in *information agents* (sometimes also called *middle agents* [4]) which are involved in the following tasks:

- finding, selecting, and querying relevant sources;
- managing and possibly processing the retrieved information; and
- updating their own knowledge about other agent services and features.

In this context, different scenarios—which to some extent overlap—have been suggested and implemented. Following [15], these can be classified as follows:

Facilitators: Agents which take control over a set of subordinated agents, coordinating the services they offer and the use of the resources they require.

Brokers: Agents often used for matching between a set of different data sources and user requests. Brokers receive requests, look for relevant sources matching these requests, and then perform actions using services from other agents (combining them with their own resources or information).

Mediators: In the mediator approach [25], meta-knowledge about a number of other agents (sometimes called *provider agents*) is available to the mediator, which exploits this knowledge to create higher-level services (not provided by the underlying agents) for user applications. These new services result by the combination and merging of low-level services on the basis of the comprehensive meta-information which the mediator has. In a sense, mediators may be seen as enhanced, high-level brokers.

Yellow Pages: A yellow pages dictionary helps users and other agents in finding the right providers (other agents or information sources) for the kind of service they need, possibly performing a match between advertised services, an ontology of the domain in question, and service requests.

Blackboards: A blackboard serves as a temporal repository for service requests which remain to be processed. Agents offering some service can access the blackboard and look for (and retrieve from it) service requests which they can satisfy.

Note that in a multi-agent system several of these agents with different features may be available at the same time, performing different tasks.

In many multi-agent systems, however, agents have only limited (if any) reasoning capabilities. These capabilities are in particular needed during decision-making processes, e.g., when an agent has to determine which actions are appropriate in response to a given stimulus. However, advanced reasoning capabilities require that knowledge—which serves as the input of any reasoning process—is represented in the agent's knowledge base in an appropriate way. In particular, aspects of advanced knowledge representation such as incomplete information, inconsistent information, or preferences need to be dealt with in an explicit, declarative way in the agent's knowledge base.

3 Activities and Projects

The activities of the Knowledge-Based Systems Group at TU Wien in this field focus on the development of advanced (reasoning) methods for better, i.e., more intelligent, information services. In particular, we are interested in employing *answer set programming* (*ASP*) [20] techniques for a number of reasons.

ASP is a recent development in nonmonotonic reasoning which uses *extended logic programs* (*ELP*s) for declarative problem solving. Besides pure declarativity, i.e., no need for paying much attention to syntactical matters, ASP is capable of handling incomplete information and performing defeasible, i.e., nonmonotonic reasoning—undoubtedly very eligible aspects for the problem domain at hand. Furthermore, methods for expressing priorities in logic programs can be applied to formally deal with (user-)preferences in a declarative way. Finally, powerful ASP tools exist, which can serve as reasoning engines in implementations. In particular, we make use of the `DLV` system [8, 12] and its `PLP` front-end [6, 5] for prioritized logic programs.

For eventually realizing advanced Web services making use of these reasoning capabilities, we "agentize" our approaches in the *IMPACT* environment [21], which is being developed as a platform for creating and deploying software agents in an open agent environment. In the *IMPACT* system, a society of software agents is maintained by agents which are registered at designed *IMPACT* servers. Each software agent which wants to be a member of the society may join by registering at some *IMPACT* server, by declaring the services it provides in a uniform service description language (SDL). The registered agent must be able to understand and process requests that arrive through this interface, following a specified protocol.

In the following, we present two projects aimed at providing intelligent Web services by means of information agents resorting to advanced reasoning capabilities.

3.1 Site Selection

In our project "Knowledge-Based Agents for Advanced Information Access" [2], we considered a particular task for information agents, namely information source selection, and developed an intelligent site selection agent building on declarative methods. Here, the problem is, given a query by the user, which out of a collection of information sources should the agent select for answering the query, such that the utility of the answer, in terms of quality of the result and other criteria (e.g., costs), is as large as possible for the user? Obviously, a sensible solution to this problem is nontrivial and involves various aspects such as basic properties of the information sources, knowledge about their contents, and knowledge about the particular application domain. These aspects have to be suitably combined, and reasoning is needed to elicit implicit knowledge.

There are several reasons for tackling the problem using a knowledge-based approach, and in particular an ASP approach, rather than a standard decision theoretic approach based on utility functions. First, the site-selection process is assigned a declarative semantics, hence a clear formal semantics, under both qualitative as well as quantitative criteria. Second, the formalism is capable of handling incomplete information and performing nonmonotonic inferences, which, arguably, is an inherent feature of the current problem domain. Third, changes in the specification of the site-selection process are easily incorporated by modifying or adding suitable rules or constraints, without the need for re-designing the given program, as might be the case, e.g., in procedural languages. Finally, the declarative nature of the answer set semantics formalism permits also a coupling with sophisticated ontology tools, and with reasoning engines for them, providing advanced features for the domain knowledge.

Our approach to information site selection can briefly be summarized as follows:

- We use extended logic programs to represent rich descriptions of the information sources, an underlying domain theory, and queries in a formal language. We perform query analysis by ASP and compute query abstractions. We consider XML-QL [7], but other formal query languages could be handled as well.

- At the heart, a declarative site selection program represents both qualitative and quantitative criteria (e.g., site preference and costs). Its rules may access information supplied by the other programs, including object and value occurrences in the query. For example, a rule r_1 may state that a query about actor awards should be posed to source s_1.

- Structured data items require a careful definition of the selection semantics. If, besides r_1 as above, a rule r_2 says that source s_2 is in general best for awards, then r_1 should have priority over r_2 and s_1 should be queried. Note that this priority is not based on inheritance, but on the context of the occurrence of awards, which is given by the query access path. Thus, inheritance-based approaches such as [3] do not apply. Furthermore, implicit priorities, derived from context information, must be combined with explicit user preferences from the selection policy, and arising conflicts must be resolved.

We have carried out experiments in the movie domain: Our example application comprises several XML databases, wrapped from movie databases on the Web, and handles queries in XML-QL. It turned out that the system behaved intuitively on a number of natural queries, some of which require reasoning from the background knowledge to identify the proper selection.

An interesting issue for further research in this context is to automatically generate site descriptions from annotated sources. Further details can be found in [9, 14, 11].

3.2 INFOMIX

The main goal of the INFOMIX project [1] is to provide a set of techniques and associated tools for powerful information integration by using advanced reasoning capabilities. It aims at developing a theory, comprising a comprehensive information model and information integration algorithms, and a prototype implementation of a knowledge-based system for advanced information integration, by using computational logic and integrating research results on data acquisition and transformation.

The objectives of the project are detailed as follows:

Comprehensive Information Model: A comprehensive information model has to be provided, which incorporates static and dynamic aspects of information integration, and supports advanced "human-like" reasoning, based on a rich semantics.

Information Integration Algorithms: A host of efficient algorithms for information integration must be provided, which can be applied to homogenized data from heterogeneous data sources.

Usage of Computational Logic: Advanced methodologies and techniques from computational logic as a toolbox for information integration ought to be exploited.

Integration of Results on Data Acquisition and Transformation: Selected methods from the area of data source acquisition and transformation should be integrated.

Prototype System: Definition and implementation of a component-based integration system prototype, and providing an infrastructure by using software agent technology.

A user should conceive the integrated information as a global database adhering to a global schema, whilst the data stems from a set of different data sources (of different formats) conforming to according local schemata. In order to answer user queries in this setting, hard reasoning tasks have to be solved, where ASP techniques apply. Fur further information about INFOMIX, see [17].

References

[1] INFOMIX: Knowledge-Based Information Integration. `http://www.kr.tuwien.ac.at/research/infomix.html`.

[2] Knowledge-Based Agents for Advanced Information Access. `http://www.kr.tuwien.ac.at/staff/giuliana/project.html`.

[3] F. Buccafurri, N. Leone, and P. Rullo. Stable Models and their Computation for Logic Programming with Inheritance and True Negation. *Journal of Logic Programming*, 27(1):5–43, 1996.

[4] K. Decker, K. Sycara, and M. Williamson. Middle-Agents for the Internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI 1997)*, volume 1, pages 578–583. Morgan Kaufmann, 1997.

[5] J. Delgrande, T. Schaub, and H. Tompits. plp: A Generic Compiler for Ordered Logic Programs. In T. Eiter, W. Faber, and M. Truszczyński, editors, *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2001)*, pages 411–415. Springer, 2001.

[6] J. Delgrande, T. Schaub, and H. Tompits. A Framework for Compiling Preferences in Logic Programs. *Theory and Practice of Logic Programming*, 2003. To appear.

[7] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu. A Query Language for XML. In *Proceedings of the 8th International World Wide Web Conference (WWW8)*, 1999. Computer Networks 31(11–16):1155–1169, See also `http://www.w3.org/TR/NOTE-xml-ql/`.

[8] T. Eiter, W. Faber, N. Leone, and G. Pfeifer. Declarative Problem-Solving Using the DLV System. In J. Minker, editor, *Logic-Based Artificial Intelligence*, pages 79–103. Kluwer Academic Publishers, 2000.

[9] T. Eiter, M. Fink, G. Sabbatini, and H. Tompits. A Generic Approach for Knowledge-Based Information Site Selection. In *Proceedings 8th International Conference on Principles of Knowledge Representation and Reasoning* (*KR 2002*)*, Toulouse, France*, pages 459–469. Morgan Kaufmann, 2002.

[10] T. Eiter, M. Fink, G. Sabbatini, and H. Tompits. Using Methods of Declarative Logic Programming for Intelligent Information Agents. *Theory and Practice of Logic Programming*, 2(6):645–719, 2002.

[11] T. Eiter, M. Fink, and H. Tompits. A Knowledge-Based Approach to Information Site Selection. Technical Report INFSYS RR-1843-02-09, Institut für Informationssysteme, Technische Universität Wien, A-1040 Vienna, Austria, July 2002.

[12] T. Eiter, N. Leone, C. Mateis, G. Pfeifer, and F. Scarcello. A Deductive System for Nonmonotonic Reasoning. In *Proceedings of the 4th International Conference on Logic Programming and Nonmonotonic Reasoning* (*LPNMR 1997*), volume 1265 of *LNAI*, pages 363–374, 1997.

[13] D. Fensel, C. Bussler, and A. Maedche. Semantic Web Enabled Web Services. In *Proceedings of the First International Semantic Web Conference* (*ISWC-2002*)*, Sardinia*, 2002.

[14] M. Fink. *Declarative Logic-Programming Components for Information Agents.* PhD thesis, Institut für Informationssysteme, Technische Universität Wien, A-1040 Vienna, Austria, 2002.

[15] R. Flores-Mendez. Towards a Standardization of Multi-Agent System Frameworks. *ACM Crossroads*, 5(4), 1999.

[16] D. Lembo, M. Lenzerini, and R. Rosati. Review on Models and Systems for Information Integration. INFOMIX Deliverable D1.1; see also `http://sv.mat.unical.it/infomix/`, 2002.

[17] N. Leone. INFOMIX: Boosting the Information Integration. Contract No.: IST-2001-33570, Annex 1 – "Description of Work"; see also `http://sv.mat.unical.it/infomix/`, 2002.

[18] A. Levy and D. Weld. Intelligent Internet Systems. *Artificial Intelligence*, 118(1–2):1–14, 2000.

[19] F. Sadri and F. Toni. Computational Logic and Multi-Agent Systems: a Roadmap. *Computational Logic, Special Issue on the Future Technological Roadmap of Compulog-Net*, 2000. `http://www.compulog.org/net/Forum/Supportdocs.html`.

[20] T.C. Son and J. Lobo. Reasoning about Policies using Logic Programs. In A. Provetti and T.C. Son, editors, *Proceedings AAAI 2001 Spring Symposium on Answer Set Programming: Towards Efficient and Scalable Knowledge Representation and Reasoning, Stanford, CA* (*Workshop Technical Report SS-01-01*), pages 210–216. AAAI Press, 2001.

[21] V.S. Subrahmanian, P. Bonatti, J. Dix, T. Eiter, S. Kraus, F. Ozcan, and R. Ross. *Heterogeneous Agent Systems.* MIT Press, 2000.

[22] W3C. Simple object access protocol (SOAP) version 1.1. 2000. `http://www.w3.org/TR/SOAP`.

[23] W3C. The UDDI Technical White Paper, 2000. `http://www.uddi.org/`.

[24] W3C. Web services description language (WSDL) version 1.2, 2002. `http://www.w3.org/TR/wsdl12/`.

[25] G. Wiederhold. Mediators in the Architecture of Future Information Systems. *IEEE Computer*, 25(3):38–49, 1992.