

Integration of Heterogeneous Sources: Towards a Framework for comparing Techniques

Valentina A.M. Tamma and Pepijn R.S. Visser

CORAL - Conceptualisation and Ontology Research at Liverpool

Department of Computer Science, University of Liverpool

P.O. Box 147, Liverpool, L69 7ZF, United Kingdom

E-mail: <valli, pepijn@csc.liv.ac.uk>

Phone: (+44) – 151 – 794 3709 Fax: (+44) – 151 – 794 3715

1. Introduction

The Internet enabled us to access huge amounts of information from different places all over the world. Stimulated by the growth of the Internet there is a growing demand for understanding how to integrate multiple and heterogeneous knowledge sources. Research on the integration of heterogeneous sources aims at recognising and combining relevant knowledge so as to provide a richer understanding of a particular domain. The integration is particularly valuable if it enables the communication between different sources while allows them to maintain their autonomy.

The vast and diverse nature of the contributions to the integration field has given rise to a large amount of literature that is not always fully accessible. The aim of this paper is to present a framework for comparing different techniques for the integration.

This aim differs from previous literature review work in that in our approach emphasis is put on the integration and the use of shared ontologies to achieve this purpose. Existing literature reviews focus on the design of ontologies (Friedman Noy & Hafner, 1997), and the comparison of methodologies for ontology development (Jones *et al.*, 1998).

This comparison framework is composed by a set of question that have been developed as part of a PhD research project addressing the problem of creating shared ontologies in distributed systems. In the questions, emphasis is given to ontologies, which according to Guarino and Giaretta (Guarino and Giaretta, 1995) can be defined as “an explicit, partial account of a conceptualisation”. Ontologies play a pivotal role in the different approaches because they can be used to express the agreement (*ontological commitment*) on the conceptualisation among heterogeneous sources willing to communicate.

This paper is organised as follows. Section 2 describes three integration projects. Section 3 discusses the differences between the projects. In section 4 the questions are the framework is applied to the projects while section 5 presents a discussion on the application of the framework and the conclusions.

2. Three integration projects

There are many projects that can be discussed to illustrate the framework, here we focus on three of them: InfoSleuth, KRAFT, and OBSERVER.

InfoSleuth (Bayardo *et al.*, 1997)

InfoSleuth is a system for the integration of heterogeneous sources developed by MCC (Microelectronics and Computer Technology Corporation, Austin, Texas, USA).

The purpose of the InfoSleuth project is to retrieve and process information in a network of heterogeneous information sources (also called *resources*).

In InfoSleuth, the heterogeneity concerns three issues: the paradigms used to represent the knowledge (also referred to as schema heterogeneity); the languages used to represent the knowledge and the conceptualisation underlying the schema. The different sources are integrated in a dynamic way and this is made possible by using a network of co-operating agents that form the InfoSleuth architecture.

The InfoSleuth architecture includes both core and application dependent components. Core application provide fundamental services, they are:

User Agent: This agent allows the user to access the InfoSleuth system. It obtains information about the ontologies known to the system and it uses them to prompt its user in selecting an ontology that will be used to formulate queries. Each of these is sent to the most appropriate task execution agent (see below) that will send the obtained results to the user agent.

Resource Agent: This agent allows the InfoSleuth architecture to access the information sources and executes the requests concerning a specific resource.

The resource agent answers queries translating them from the common query language into a language understood by the resources. This translation comprises both the mapping of the shared ontology into database schema, and the mapping of the query language into the native language.

Ontology Agent: This agent is a specialised Resource Agent whose main task is to answer questions about ontologies. It answers queries about the ontologies available, such as the source of an ontology and searches the ontologies for concepts.

Broker Agent: This agent aims at finding the resources required to solve a user query. All InfoSleuth agents advertise their capabilities to the broker agent that semantically matches agents looking for a particular service with agents providing that particular service (*information brokering technique*).

At least, an agent has to advertise its name, its location and its language, but it can also advertise meta-information and domain constraints. The advertisement is expressed in terms of one or more ontologies thus enabling the dynamic matching.

Task Execution Agent: This agent routes requests to the appropriate Resource Agents. It decomposes user queries into sub-queries and reassembles the answers, thus co-ordinating the executions of high-level information gathering sub-tasks. The strategy followed is based on task plans with procedural attachments.

The application dependent components of the InfoSleuth architecture contribute only to some applications. They are:

Data Analysis Agent: This agent performs data analysis/mining operations.

Monitor Agent: This agent stores records of the agent interactions and of the task execution steps.

The co-operation between multiple agents is obtained by using the information brokerage technique that routes all the requests only on to the relevant resources. Information brokerage and ontologies are two aspects of the InfoSleuth approach strictly intertwined. Agent communications take advantage of the use of ontologies as they are used to the agent infrastructure (this is done by specifying the information and the relationships between the various agents). This aids the routing of the requests to a specific agent.

InfoSleuth allows different formats and representations of ontologies by the use of an ontology meta-model that provides a unified view on the way ontologies are specified. In this way agents might reason about ontologies using different languages depending on the type of inference to be made.

KRAFT (Gray *et al.*, 1997)

KRAFT (Knowledge Reuse and Fusion / Transformation) is a multi-site research project conducted at the universities of Aberdeen, Cardiff and Liverpool in collaboration with BT (British Telecommunications PLC) in the UK.

The overall aim of this project is to enable the sharing and reuse of constraints embedded in heterogeneous databases and knowledge systems. In the KRAFT approach to the integration problem there are three types of heterogeneity: ontological assumptions (conceptualisations and organisations of the data), paradigm and language.

KRAFT recognises a small number of shared ontologies. Moreover each resource has its own local ontology, and provides a translation to at least one shared ontology; in this way local ontologies allow the communication between heterogeneous resources that can maintain their intrinsic heterogeneity.

The KRAFT network has the following components:

User Agent: is the interface between users and services provided by KRAFT domain;

Resource: is the knowledge source to integrate. It provides services to the KRAFT domain. Examples of KRAFT resources are databases, knowledge bases and constraint solvers.

Wrapper: is the interface between the domain and the user agent or the resources. Wrappers provide communication services, both at high and at low level. At high level they support the mechanisms linking the resources to mediators and facilitators (see below). At low level they provide a translation service between the internal data formats of users agent and resources and the internal data format supported by the KRAFT domain. They co-operate with the ontology agent (see below) to perform translations.

Mediator: is the component that retrieves information on a domain. In achieving this purpose it uses domain knowledge to transform data. It performs operations on queries to implement a certain task and can process queries by decomposing, combining them and transforming their content.

Ontology Agent: is the component that translates knowledge expressed against a source ontology into the knowledge expressed against a target ontology. If a mediator or a wrapper requires an ontology translation it passes the expression and references to both source and target ontologies to the ontology agent who will translate and return the expression.

Facilitator: is the KRAFT component performing the internal routing services for messages within the KRAFT domain. Its main functions are to maintain records of the location and of the capabilities of the resources, and to accept and route messages from other KRAFT resources.

OBSERVER (Mena *et al.*, 1998)

OBSERVER (Ontology Based System Enhanced with Relationships for Vocabulary hEterogeneity Resolution) is a project presently conducted at the University of Zaragoza, Spain.

The aim of the OBSERVER project is to retrieve and process information stored in heterogeneous knowledge sources (called repositories). The heterogeneity in this project concerns paradigms and ontological assumptions. To overcome the differences in the formats and in the languages OBSERVER relates repositories to domain ontologies; these are pre-existing ontologies defining a set of terms in a specific domain. The OBSERVER (Goñi *et al.*, 1997) architecture comprises four main components:

Query processor: This component has as input a user query expressed in a chosen *user ontology*. The query processor accesses the data repositories to answer the query. If the user is not satisfied with the answer, the query processor translates (partially or totally) the

query into another user-selected ontology using predefined inter-ontology relationships. The query processor generates a list of translation plans, where each plan has an associated loss of information.

Ontology server: This component provides the user processor with mappings that link each term in an ontology with structures in data repositories and it translates queries for the retrieval of data from the repositories. In the access the ontology server is assisted by the wrapper (see below) of the corresponding data repository.

Interontology Relationships Manager (IRM): This component deals with inter-ontology relationships that relate terms in different ontologies. OBSERVER considers three kinds of possible relationships: *synonym, hypernym and hyponym*.

Wrapper: This component has knowledge of the data organisation in the repositories. The wrapper actually accesses the data repository using the mapped information provided by the ontology server.

The processing is performed according to the following steps: First users choose one domain ontology whose term will be used to build the query. Once the query is formulated, the ontology server verifies its syntax, then it performs ontological transformations of the query, and decomposes it. After the decomposition, the ontology server uses relevant mappings rules to relate terms in the ontology to the data structure in the underlying repositories. In accessing the repository to retrieve a queried data, the ontology server is assisted by the wrapper. Once the data is retrieved, the ontology server returns the user with the answers obtained. If the user is not satisfied with the answer, the query processor reformulates the query using another user chosen ontology.

3. Towards the comparison framework

In this section we discuss the most relevant questions of our framework. To clarify the aim of the questions the three above described projects are considered.

The projects differ in the sources they integrate. Some projects address the problem of the integration of databases and knowledge bases. There are also projects devoted to integrate only databases but of different paradigms, or concerning specifically the integration of different databases of the same paradigm (e.g. relational, object-oriented). Other projects extend the integration to other kinds of knowledge sources, such as HTML pages, images or textual data. Question 1 of the classification framework addresses this issue.

As can be seen from the descriptions of the projects, the heterogeneity may concern several aspects of a system. In our framework the following classification is adopted (Visser *et al.* 1998):

Content heterogeneity: if two systems represent different knowledge;

Paradigm heterogeneity: if different knowledge sources express knowledge by means of different modelling paradigm;

Language heterogeneity: if knowledge sources express their knowledge using different representation languages;

Ontological heterogeneity: occurs when different systems use different conceptualisations. Conceptualisations can differ because they describe different viewpoints about a domain or they can describe the world from the same viewpoint but one system can include in its conceptualisation more concepts than another.

These types of heterogeneity are relatively independent from each other, so even if two systems represent knowledge by using the same paradigm, in the same language, and with the same ontological assumptions, if the content is different they are to be considered heterogeneous. The type of the heterogeneity is addressed in question 2.

We have already stated that integrating knowledge means to agree on ontological commitments. This agreement can be implicit or explicit. For example, KRAFT and InfoSleuth exchange information via an intermediate ontology while OBSERVER integrates the sources in a more direct way, without using an intermediate ontology. OBSERVER defines relationships that relate terms in one ontology with terms in another one.

Possibly, more than one explicit shared ontology is used for the integration. In particular, a system can be classified in one of the following classes: a) No shared ontology (there is a direct translation between one source and another); b) One shared ontology (one ontology is used to integrate the various sources and several mapping functions are defined between the resource and the shared ontology); c) Multiple shared ontologies (the integration is accomplished by using multiple shared ontologies and there are mapping functions both between the resource and the shared ontology and between the different ontologies, as in KRAFT and InfoSleuth). How many explicit shared ontologies are used for the integration is addressed in question 16.

If there are multiple ontologies, the projects can differ in the principles used to obtain these ontologies. There are several different partitioning principles (Visser & Cui, 1998), such as domain partitioning, alternative domain views, abstraction, primary ontologies versus secondary ontologies, terminological, informational and knowledge modelling ontologies, meta-level descriptions (question 18). Projects can also differ in the relationships (Visser & Cui, 1998) linking these multiple ontologies, such as subset/superset relation,

extension relation, restriction relation, mapping relation (question 19).

There are also differences in the type of knowledge considered in the integration. Some systems, for example KRAFT, integrate constraint-based knowledge. This kind of knowledge is richer than mere data but it requires more complex architectures (question 8).

Even if all the projects described use ontologies to integrate knowledge, they differ in the way in which they use them. Some projects use existing ontologies such as PENMAN (Bateman *et al.*, 1990) and WordNet (Miller *et al.*, 1993) (question 6). This approach is followed in the OBSERVER project and in KRAFT (although KRAFT builds its own ontologies starting from a pre-existing one). In other projects the ontologies are developed from scratch, such as InfoSleuth.

The organisation of the ontologies is an important classification feature (question 12). For example, do the projects organise their ontologies as a hierarchy and, if so, how many hierarchies do they distinguish? Alternatively, are their ontologies organised as a lattice, or as a graph?

4. The comparison framework

This framework consists of 19 questions divided into 3 groups: The first one, (A) *general architectural features*, focuses on the projects architectural aspects; in the second, (B) *Intra-ontology features*, emphasis is put on the internal ontology structure, while the third group, (C) *Inter-ontology features*, investigates the relationships between multiple ontologies. (cf. Visser and Bench-Capon, 1998).

A. General architectural features

1. What are the heterogeneous sources that are integrated? See section 3.
2. What kinds of heterogeneity are distinguished? See section 3.
3. Is the integration task performed using an agent architecture?
4. Does the proposed solution address the problem of adding/removing new resources?
5. Has the approach been implemented (that is, is it operational) and is it available?

B. Intra-ontology features

6. Does the system use pre-existing ontologies or does it build its own ontologies concerning a domain? See section 3
7. Does the proposed solution address the problem of an evolving ontology?
8. Does the ontology have constraints? See section 3
9. What is the ontology-specification language (e.g. Ontolingua, KIF, DL, Prolog, Loom, Horn clauses)?

10. To what extent is the inference capability of the language used (e.g. not at all, subsumption)?

11. Does the approach distinguish an explicit meta-level ontology (e.g. frame ontology)?

12. How is the content of the ontologies organised? See section 3

13. If there is a hierarchy, what top-level ontology (such as the empty ontology, WordNet, PENMAN/PANGLOSS, a unified top-level ontology) is used? For top-level ontologies we refer to (Sowa, 1995) and (Guarino, 1997)

14. What are the relationships within the ontologies (e.g. Is-a relationship, Part-of relationship)?

C. Inter-ontology features

15. How many semantic translation steps are performed (e.g. none, one, two, more than two)?

16. How many shared ontologies are distinguished?

17. Does the multiple ontology structure allow multiple parents?

18. If there are multiple ontologies, what is the partitioning principle? See section 3

19. What are the relationships between multiple ontologies? See section 3

The framework is used to compare three projects InfoSleuth, KRAFT, and OBSERVER and the table in Figure 1 illustrates results of the comparison.

5. Discussion and conclusions

The discussion in section 3 illustrates the need for a comparison framework of techniques in the literature. Sometimes two projects seem to have common aspects at first glance. KRAFT and OBSERVER, for instance, distinguish a component called *wrapper*. In OBSERVER the wrapper seems to be confined to providing access to the data repositories whereas in KRAFT it seems to have a more sophisticated role, in that provides communication services at different levels. Moreover, in contrast to OBSERVER, in KRAFT it is also used to interface with the user.

In other cases the similarity between techniques is not really evident at a first glance. This is the case in KRAFT and InfoSleuth. Despite differences in the naming of components there are substantial similarities in the functions performed. For example, the functions that in InfoSleuth are performed by the broker agent in KRAFT are performed by the facilitator. Moreover, both projects have an ontology agent, although the KRAFT ontology agent seems to have more capabilities than its counterpart in InfoSleuth. Finally, with respect to internal routing there is a strong similarity between the functionality of a mediator in KRAFT and that of a task execution agent in InfoSleuth.

Acknowledgements

The authors would like to thank Ian Finch and Michael Shave for their contribution to this paper. The research

presented in this paper is funded by BT. The authors are grateful to Zhan Cui and Paul O'Brien for their support.

Referenecces

Bateman, J. A. *et al.*, 1990: A General Organization of Knowledge for Natural Language Processing: the PENMAN Upper Model. Technical report, USC/Information Sciences Institute, Marina del Rey, California, USA.

Bayardo, R.J. *et al.*, 1997: InfoSleuth: Agent-Based Semantic Integration of Information in Open and Dynamic Environments. In *ACM SIGMOD Record Vol. 26, No. 2 (June 1997)*, *SIGMOD '97. Proceedings ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, USA, pp. 195-206

Fridman Noy, N., and Hafner, C.D., 1997: The State of the Art in Ontology Design: A Survey and Comparative Review. In *AI magazine*, 18, 3, pp. 53-74.

Goñi, A. *et al.*, 1997: Querying Heterogeneous and Distributed Data Repositories using Ontologies. In *7th European-Japanese Conference on Information Modelling and Knowledge Bases (IMKB '97)*, Toulouse, France.

Gray, P.D.M. *et al.*, 1997: KRAFT: Knowledge Fusion from Distributed Databases and Knowledge Bases. In *Proceedings of the DEXA 1997*, Toulouse, France, pp. 682-691.

Guarino N., and Giaretta, P., 1995: Ontologies and Knowledge Bases: Towards a Terminological Clarification. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, Mars, N. (Ed.), IOS press, Amsterdam, pp. 25-32.

Guarino N., 1997: Some Organising Principles For A Unified Top-Level Ontology. In *Working Notes of AAAI Spring Symposium on Ontological Engineering, Stanford, USA*. Revised version LADSEB-CNR Int. Rep. 02/97.

Jones, D.M. *et al.*, 1998: Methodologies for Ontology Development. In *Proceedings of IT&KNOWS Conference of the 15th IFIP World Computer Congress*, Chapman-Hall, Budapest, Hungary.

Mena, E. *et al.*, 1998: Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. In *Proceedings of the International Conference On Formal Ontology In Information Systems (FOIS '98)*, Trento, Italy.

Miller, G.A. *et al.*, 1993: Introduction to WordNet: AnOn-line Lexical Database. *Five Papers on WordNet*, CSL Report, 43 Cognitive Science Laboratory, Princeton University, USA, July 1990, revised August 1993.

Sowa, J.F., 1995: Top-level Ontological Categories. In *International Journal of Human-Computer Studies*, 43, 5-6, pp.669-685.

Visser, P.R.S., and Bench-Capon, T.J.M., 1998: Ontologies in Legal Information Systems. In *Proceedings of the Marvin Farber Conference on Applied Ontologies*, Buffalo, NY, USA, pp. 76-85.

Visser, P.R.S., and Cui, Z., 1998: Heterogeneous Ontology Structures for Distributed Architectures. In *Proceedings of the ECAI '98 Workshop on Applications of Ontologies and Problem-Solving Methods*, Brighton, UK.

Visser, P.R.S. *et al.*, 1998: Assessing Heterogeneity to Classify Ontology Mismatches. In *Proceedings of the International Conference On Formal Ontology In Information Systems (FOIS '98)*, Trento, Italy.

Qn.	KRAFT	OBSERVER	InfoSleuth
1.	Not restricted	Relational and object- oriented databases, HTML pages	Relational databases, object-oriented databases, textual data, HTML pages.
2.	Paradigm.and Ontological heterogeneity	Paradigm and Ontological heterogeneity	Paradigm, Language, and Ontological heterogeneity
3.	Yes	No	Yes
4.	Yes	Yes	Yes
5.	Partially implemented and not freely available.	Partially implemented and freely available.	Partially implemented and not freely available.
6.	Builds its own ontologies starting from WordNet	Uses only pre-existing ontologies	Builds its own ontologies.
7.	No	No	No
8.	Yes	Ontologies are use to constrain user queries	Yes, in a certain way it does.
9.	P/FDM	DL	KIF and LDL++
10.	Constraint check, inheritance	Deductive reasoning capabilities	LDL deductive mechanism help in the consistency check of the query constraints
11.	Yes	No	Yes
12.	Hierarchies	Lattice	There is no explicit hierarchy
13.	WordNet	There is not a top-level ontology	There is not a top-level ontology
14.	Is-a relationship	Is-a relationship	Unknown
15.	Two	At least one	Two or more semantic translations.
16.	At least one	None	Multiple mappings
17.	Yes	No	Unknown
18.	Alternative domain views, domain partitioning and abstraction	Alternative domain views	Abstraction, alternative domain views, meta-level description.
19.	Extension, superset relationships	Mapping relationships	Mapping relationships

Figure 1 the classification framework applied to three projects