EnviroInfo 2001: Sustainability in the Information Society
Copyright 2001 Metropolis Verlag, Marburg, ISBN: 3-89518-370-9

Intelligent Brokering of Environmental Information with the BUSTER System

H. Neumann, G. Schuster, H. Stuckenschmidt, U. Visser,
T. Vögele and H. Wache[1]

Abstract

In this paper we discuss the general problems which arise when information sources have to be found and integrated into a system and present a solution by introducing the BUSTER[2] system. We give an overview of our ontology-based approach with logical reasoning on metadata for retrieving information sources and semantic translation into the desired format.

1. Motivation

Many application areas of information systems share the need to store and process large amounts of diverse data which are often geographically distributed. This implies that making new data available to the system requires the data to be transferred into the system's specific data format. This is a process which is very time-consuming and tedious. Data acquisition, automatically or semi-automatically, often makes large-scale investment in technical infrastructure and/or manpower inevitable. These obstacles are some of the reasons behind the concept of intelligent brokering of information. Our approach can be applied because existing information can be accessed by remote systems in order to supplement their own data basis. The advantages of successful brokering are thus obvious for many reasons:

- Quality improvement of data owing to the availability of large and complete data.
- Improvement of existing analyses and application of the new analyses by the use of more relevant data sources.
- Cost reduction resulting from multiple use of the existing information sources.
- Avoidance of redundant data and conflicts which may arise from redundancy.

---

[1] Intelligent Systems Group, Center for Computing Technologies, University of Bremen,
e-mail: {neumann, schuster, heiner, visser, vogele, wache}@tzi.de

[2] Bremen University Semantic Translator for Enhanced Retrieval

However, in order to establish efficient information brokering, difficulties arising from organizational and competence questions and many other technical problems have to be solved:

1. A suitable information source which contains the data needed for a given task must be located.
2. Once the information source has been found, access to the data therein has to be provided.
3. Furthermore, access has to be provided on a technical and informational level.

In short, intelligent brokering does not only need to provide full accessibility to the data, it also requires that the accessed data may be interpreted by the remote system. While the problem of providing access to information has been largely solved by the invention of large-scale computer networks, the problem of processing and interpreting retrieved information remains an important research topic (Visser/Stuckenschmidt/Wache/Vögele 2001).

In section two of this paper we present the basic approach of the BUSTER system and substantiate the idea in the following chapter. Section four clarifies the functioning of the system by passing through all steps in a sample query. Chapter five finally outlines some unsolved problems and next steps for our work.

2. The BUSTER Approach

In systems with a large number of available data sources, it is often not trivial to find the right set of data for a given task. If, for example, an information request is submitted to an information broker, the broker has to decide which of the registered sources it should use to answer the request. The BUSTER approach addresses the above mentioned questions by providing a common interface to heterogeneous information sources in terms of an intelligent information broker. A user can submit a query request to the network of integrated data sources. In this query phase several components of different levels interact.

The first step is to find the desired information source. To do this, BUSTER uses a certain ontology, the lookup ontology, which is built according to a metadata base. Metadata, i.e. data describing a data source, are often used to organize and manage large collections of data sources. Typically, such metadata catalogues are based on standardized metadata formats such as the Dublin Core metadata format (Hillman 2001).

In a second step user queries are matched against this ontology. If the matching succeeds, the broker establishes a connection to the actual information source. If the matching fails, the broker decides that there is no valuable information available.

Otherwise, in the last step of the BUSTER approach, the semantic translation is started. Each data source is represented by a specific ontology, the so-called source ontology. It contains an explicit description of the concepts covered by the data source. In addition, it contains information about the structural and syntactic details of the data source.

All these ontologies (especially the source ontologies) have to use the same vocabulary describing the concepts and the metadata concerning the data source.

We will now take a closer look at the conceptional organization of the BUSTER system and outline three levels of interacting components (compare figure 1).

On the syntactic level, wrappers are used to establish a communication channel to the data source(s) which have been found, that is independent of specific file formats and system implementations. Each generic wrapper covers a specific file- or data-format. For example, generic wrappers may exist for ODBC data sources, XML data files, or specific GIS formats. Still, these generic wrappers have to be configured for the specific requirements of a data source.

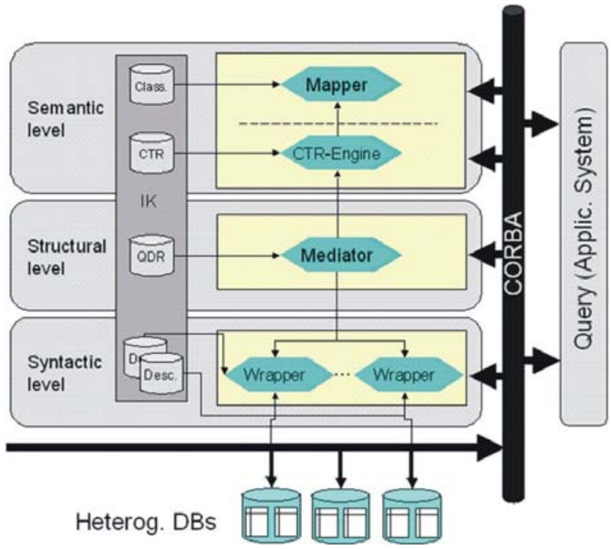

Figure 1: The BUSTER approach

The mediator on the structural level uses information obtained from the wrappers and combines, integrates, and abstracts them. BUSTER allows the use of different mediators which are configured by transformation rules. These rules describe in a declarative style how the data from several sources can be integrated and transformed to the data structure of original source.

On the semantic level, we use two different tools specialized for solving the semantic heterogeneity problems based on the source ontologies describing the contents of the information sources. Both tools – the Feature Manipulation Engine

FME, a conversion tool for geographical data formats (Safe Software Inc. 2001) and the MeCoTa Mediator (Wache 1999) – are responsible for the context transformation, i.e. transforming data from a source-context to a goal-context.

There are several ways how the context transformation can be applied. In BUSTER we consider the functional context transformation and context transformation by re-classification.

For the description of content meta-data, BUSTER uses the language OIL, which has been developed in the context of the On-To-Knowledge project (www.ontoknowledge.org) as a proposal for a language for specifying and exchanging ontologies (Fensel/Horrocks/Van Harmelen/Decker/Erdmann/Klein 2000). OIL tries to provide a core set of features which have been widely accepted to be useful. OIL combines frame-based modelling primitives, reasoning facilities from description logics, and a tight interaction with meta-data standards on the web such as RDF and XML. We use OIL to build a semantic context model of our example data by identifying a set of common properties which can be used to define a land use class.

3. The BUSTER System

A first prototype of the BUSTER approach has been implemented. The current functionality includes ontology-driven search for information sources as well as semantic integration of geographical information sources. The prototype is built upon tools which have been developed at the university of Manchester to facilitate the use of the OIL language:

- FaCT, a logical reasoning service which can be used to check ontologies for consistency and for computing subclass relations not explicitly contained in the ontology (Horrocks 1999).
- The Ontology Editor OilEd (Bechhofer/Horrocks/Goble/Stevens 2001) providing a graphical interface for the definition of complex ontologies and a direct interaction with the FaCT reasoner in a client-server architecture. The editor is used to create meta-data models as well as context definitions used in the semantic translation step.

In figure 2 we display the components of the BUSTER system from a functional point of view. A client will be able to connect to the system over the internet and activate one of the services provided. We will now concentrate on the brokering functionality.

The user is asked to restrict the defining properties of a data source in order to restrict the set of all information sources to those of interest. At the moment, the FaCT reasoner is the main inference engine of the BUSTER system. The resulting class definition is passed to the reasoner which places the query in a hierarchy of classes. Each class is a surrogate for an information source. All classes placed in the sub tree rooted at the query class are returned, because they fulfil the constraints defined in the query. The BUSTER system presents the information sources matching the query.

The information source is labelled and several services are shown. In this case, the user can now either directly view the information as an image or define a target file format the information source should be converted to. Currently, for displaying an image FME is used to create the output format. For the semantic transformation any configured mediator could be used – as a standard we use MeCoTa.

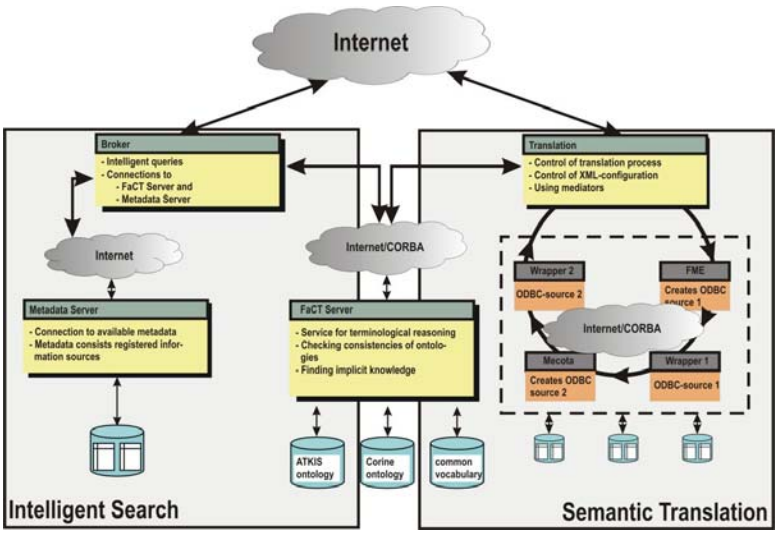

Figure 2: Client-Server architecture of the BUSTER prototype

4. A Sample Query

In the following paragraphs we will take a closer look at the system by examining all components while BUSTER is answering a query.

Assuming a user of the system has a fairly old map from the area of Bad Nenndorf (located in Lower Saxony) and does not know if the information about the land use in this area is still correct. Furthermore, the user has no knowledge about the surrounding area and is therefore looking for the missing information on the web by using BUSTER.

In a first step, the user has to select the desired type of data source. In the moment BUSTER supports just one – the data object – but we are planning to implement more. The selection of the type qualifies the properties which could be specified in the next pace.

The user now has the possibility to specify the request by choosing a term for each concept (in this case “provider”, “spatial-extend” and “topic area”). For each concept all possible entries are extracted from the lookup ontology and displayed in a choice box. Our sample user now specifies his query by building the following terms (compare figure 3):

- Non-commercial provider by choosing “commercial” and “not”– because she or he does not want to pay for the information.
- Spatial-extend should be “Lower Saxony” (by choosing “Lower Saxony” and “or”) – the location of interest is somewhere in this area, and Bad Nenndorf itself would be too specific for getting information about an area next to this district.
- “Land cover” as topic area (by choosing “Land cover” and “or”) – this results in all data sources which deal with the use of the land.

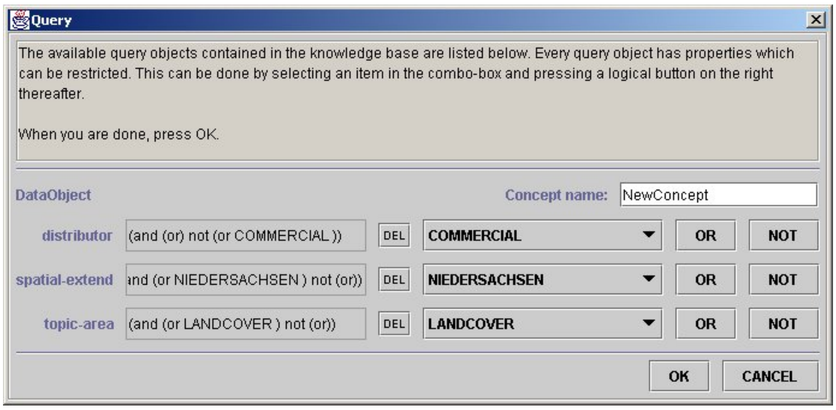

Figure 3: Specifying a query with BUSTER

At this point of time the reasoner has to look for the matching data sources. For this task this component has to classify the entered description of a data source in the lookup ontology. It will return those information sources which fulfil all restrictions and display them to the user. For this, additional information has been added from a metadatabase.

From this list the user will choose one of the presented files and the desired data format (that should be one of the formats, which could be interpreted by the users computer). In our example the retrieved data is stored in the CORINE (Bossard/ Feranec/Otahel 2000) format and the old data obtained by the user in ATKIS (AdV 1998). Because of this mismatch the user selects the context transformation “CORINE to ATKIS” – without this the retrieved information will be of restricted use (figure 4).

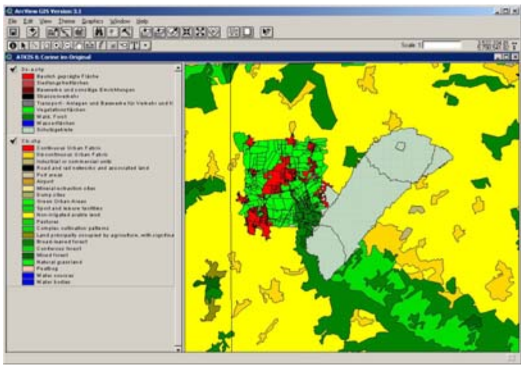

Figure 4
Before context transformation

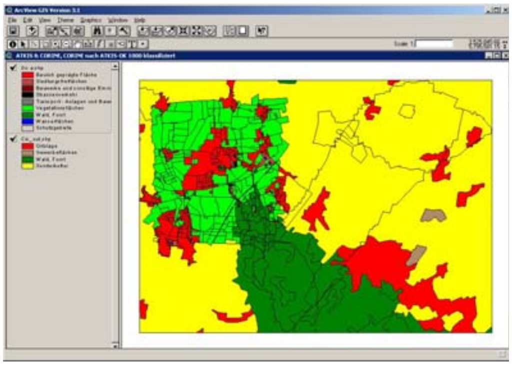

Figure 5
After context transformation

The installed mediator knows where to get the file, calls a wrapper to get the input file and a second wrapper to write the output file in the format. In between the context transformation is performed. The just created data will now be saved on the user’s computer and the process of brokering and retrieving of environmental information is completed. The user now is able to work with the new data (figure 5).

5. Future Work

In the previous chapters we discussed how the BUSTER system operates at the moment, but did not mention the unsolved problems and next steps for the system. In the last months we have outlined the following points:

- When we plan to use another data source in the system, we have to build the source ontology by hand. Is there a possibility to do this (semi-) automatically?
- When a user wants to register a certain data source in the BUSTER system, one has to describe this source in the lookup ontology. Is it possible to extract the needed metadata (semi-) automatically from the original file?
- Do we have to use a special metadata standard for the lookup ontology such as Dublin Core metadata format to facilitate the use of the system? In this case the owner of a data source has to use this metadata format, too. Is this constraint acceptable?
- We are also thinking of using another reasoner for the search for information sources and semantic translation, e.g. RACER (Haarslev/Möller 2000).

Currently we are working on a redesign of the prototype based on the use of agents. A first implementation with FIPA-OS (Poslad/Buckle/Hadingham 2000) and an ontology agent for the semantic transformation has been created and the results are promising. Further agents could capsulate the wrappers and mediators.

References

AdV (1998). Amtliches Topographisch Kartographisches Informationssystem ATKIS. Technical report, Landesvermessungsamt NRW, Bonn.

Bechhofer, S. and Horrocks, I. and Goble, C. and Stevens, R. (2001). OilEd: A Reason-able Ontology Editor for the Semantic Web, To appear in 24th German / 9th Austiran Conference on Artificial Intelligence (KI 2001).

Bossard M. and Feranec, J. and Otahel, J. (2000). The revised and supplemented CORINE land cover nomenclature. European Environment Agency, Technical Report No. 38.

Fensel, D. and Horrocks, I. and Van Harmelen, F. and Decker, S. and Erdmann, M. and Klein, M. (2000). OIL in a Nutshell. 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000), Juan-les-Pins, France, pp. 1-16.

Haarslev, V. and Möller, R. (2000). Expressive ABox Reasoning with Number Restrictions, Role Hierarchies, and Transitively Closed Roles. In: A.G. Cohn, F. Giunchiglia, and B. Selman, editors, Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning (KR2000), San Francisco, CA, 2000. Morgan Kaufmann Publishers, pp. 273-284.

Hillmann, D. I. (2001). Using Dublin Core. http://dublincore.org/documents/2001/04/12/usageguide/, 06.07.2001.

Horrocks, I. (1999). FaCT and iFaCT. Proceedings of the International Workshop on Description Logics (DL'99). P. Lambrix, A. Borgida, M. Lenzerini, R. Möller and P. Patel-Schneider, pp. 133-135.

Poslad S., Buckle P. and Hadingham R.G. (2000). The FIPA-OS agent platform: Open Source for Open Standards, Proceedings of PAAM 2000, Manchester, UK, pp. 355-368.

Safe Software Inc. (2001). Semantic Data Translation: Using FME. http://www.safe.com/semantic_links_page.htm, 06.07.2001.

Visser, U. and Stuckenschmidt, H. and Wache, H. and Vögele, T. (2001). Using Environmental Information Efficiently: Sharing Data and Knowledge from Heterogeneous Sources. Environmental Information Systems in Industry and Public Administration. C. Rautenstrauch, S. Patig. Hershey, PA, IDEA Group, pp. 41-74.

Wache, H. (1999). Towards Rule-Based Context Transformation in Mediators. International Workshop on Engineering Federated Information Systems (EFIS 99), Kühlungsborn, Germany, Infix-Verlag, pp. 107-122.