SEWASIE
Semantic Webs and AgentS in Integrated Economies
IST-2001-34825

WP7
Task T7.5
Deliverable D7.7

Showcase featuring a meaningful subset of the SEWASIE prototypes

(FINAL 1.1, 7/5/2005)

Abstract –
This deliverable describes briefly the prototype that has been developed in the SEWASIE project.

Enrico Franconi
FUB

Document information

Document ID code	D7.7		
Keywords			
Classification	**FINAL**	Date of reference	**7/5/2005**
Distribution level	**This document may be distributed to the general public for dissemination and information purposes.**		

Editor	Enrico Franconi	FUB
Authors	Matthias Jarke	RWTH
	Christoph Quix	RWTH
	Andreas Becks	FIT
	Enrico Franconi	FUB
Reviewer	Daniele Montanari	UniMORE

Version history		
Date	*Version*	*Description*
1/3/2005	DRAFT 0.9	Final draft version for internal review
20/4/2005	FINAL 1.0	Final version of the deliverable
7/5/2005	FINAL 1.1	Fixed a figure

Copyright notices

Contents

1 Executive summary

The SEWASIE project aims to design and implement an advanced search engine enabling intelligent access to heterogeneous data sources on the web via semantic enrichment. This is the basis for value-added services such as ontology based electronic negotiations or by linking semantically rich search results to OLAP reports. This deliverable describes briefly the prototype that has been developed in the SEWASIE project. The architecture will be shown following a typical business scenario, from both the users' and operational perspectives. The prototype provides users with a search client that has an easy-to-use query interface to define semantic queries. The query is executed by a sophisticated query engine that takes into account the semantic mappings between ontologies and data sources, and extracts the required information from heterogeneous sources. Finally, the result is visualised in a useful and user-friendly format, which allows identifying semantically related clusters in the documents found.

2 Introduction

SEWASIE has implemented an advanced search engine that provides intelligent access to heterogeneous data sources on the web via semantic enrichment to provide the basis of structured web-based communication. The prototype provides users with a search client that has an easy-to-use query interface to define semantic queries. The query is executed by a sophisticated query engine that takes into account the semantic mappings between ontologies and data sources, and extracts the required information from heterogeneous sources. Finally, the result is visualised in a useful and user-friendly format, which allows identifying semantically related clusters in the documents found. From an architectural point of view, the prototype is based on agent technology, i.e. the individual components of the system are implemented as agents, which are distributed over the network and communicate with each other using a standardised agent protocol.

2.1 Business Scenario

Throughout Europe, much of the industrial fabric is made of small and medium-sized enterprises (SMEs) in fields such as agriculture, manufacturing, commerce and services. For social and historical reasons, these tend to aggregate into sectorial clusters in various parts of respective countries. Today, this kind of economic organisation is threatened by globalisation.

One of the keys to sustainability and success is being able to access information. This could be a cheaper supplier, an innovative working method, a new market, potential clients, partners, sponsors, and so on. Current Internet search tools are inadequate because they not only are they difficult to use, the search results are often of little use with their pages and pages of hits.

Suppose an SME needs to find out about a topic - a product, a supplier, a fashion trend, a standard, etc. For example, a search is made for 'fabric dyeing processes' for the purpose of finding out about the disposal of the dyeing waste material. A query to www.google.com for 'fabric dyeing' listed 44.600 hits at the time of writing, which related not only manufacturers of fabric dyeing equipment, but also the history of dyeing, the dyeing technology, and so on. Eventually, a useful contact may be found, and the search can continue for relevant laws and standards concerning waste disposal. But is it law or the interpretation of the law? What if the laws are of a different country where the practices and terminologies are different?

3 SEWASIE Architecture

Figure 1 gives an overview of the architecture of the SEWASIE system. A user is able to access the system through a central *user interface* where (s)he is provided with tools for query composition, for visualising and monitoring query results, and for communicating with other business partners about search results, e.g. in electronic negotiations.

SEWASIE Information Nodes (SINodes) are mediator-based systems, providing a virtual view of the information sources managed within a SINode. The system may contain multiple SINodes, each integrating several data sources of an organisation. Within a SINode, wrappers are used to extract the data and metadata (local schemas) from the sources. The Ontology Builder is a semi-automatic tool to create an integrated ontology (the "Global Virtual

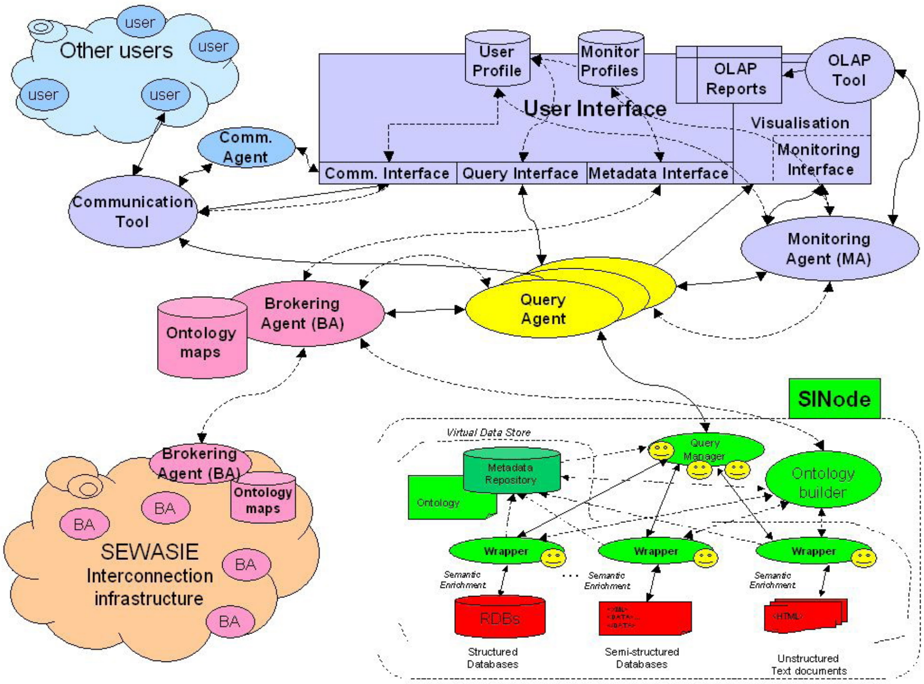

Figure 1: SEWASIE Architecture

View", GVV) from the local schemas. The GVV and the mappings to the source schemas are stored in a metadata repository. The GVV is exported as ontology to the other components of the SEWASIE architecture, and queries expressed in terms of the GVV can be processed by the *query manager* of the SINode.

Brokering Agents integrate several GVVs from different SINodes to a *Brokering Agent Ontology* (BA Ontology). The BA ontology is of central importance to the SEWASIE system. On the one hand, the user formulates the queries using this ontology. On the other hand, it is used to guide the *Query Agents* to the SINodes providing data for a query. The SEWASIE network can have multiple brokering agents, each one representing a collection of SINodes for a specific domain. Mappings between different brokering agents may be established.

A *Query Agent* receives the queries (expressed in terms of a specific BA ontology) from the user interface, rewrites the query in terms of the GVVs of the SINodes (in cooperation with the brokering agent) and sends the queries to the SINodes. The result is integrated and stored in a result repository, so that it can be used by the various end-user components.

For example, *Monitoring Agents* can be used to store a query result in a permanent repository. The monitoring agent will then execute the query repeatedly, and compare the new results with previous results. The user will be notified if a document has changed that fits her monitoring profile. Furthermore, the monitoring agent can link multidimensional OLAP reports with ontology-based information by maintaining a mapping between OLAP models and ontologies.

Finally, the *Communication Tool* provides the means for ontology-based negotiations. It uses query results, the ontologies of the Brokering Agents, and specific negotiation ontologies as the basis for a negotiation about a business contract. In addition, it uses several agents to support the negotiators in their decision process (e.g. by filter and ranking offers

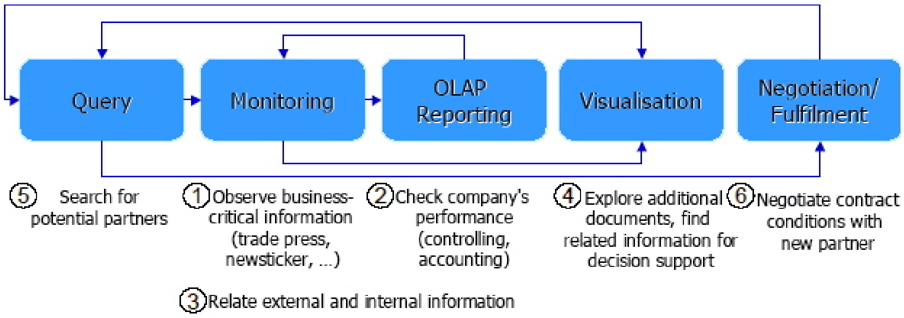

Figure 2: Demonstration scenario

of potential business partners, or by monitoring the available resources of a company).

4 Demonstration Scenario

The demonstration is based on the scenario indicated in figure 2. In the following, we will describe the system using this scenario from different perspectives: the end-user perspective (what the user sees of the system), the operational perspective (how the system works), and the design perspective (what ontology designers and administrators have to do to make the system operational). Note that also other “execution flows” are possible as it is indicated by the arrows in figure 2.

4.1 End-User Perspective

Suppose that a manager of a company starts her working day with the observation of business-critical information, such as trade press or news ticker articles (step 1). She switches to the OLAP tool to analyse the company's performance (step 2). The data of the OLAP tool represents only internal information of the company and therefore needs to be related to external information, e.g. number of sales is related to the market situation (step 3). This task is supported by the monitoring and visualisation interface (figure 3), which provides an aggregated view of documents, which have been monitored or found by a query. The tool visualises a list of documents (upper right corner) in a document map (upper left corner). Documents which are semantically close are shown in the same cluster of the map. Furthermore, the tool links the dimensions of the OLAP model (lower right corner) to the concepts of an ontology (lower left corner). Thereby, it enables direct semantic queries from the OLAP tool, i.e. find all documents related to an OLAP report. This could be used for step 4 of the scenario.

In the scenario, the user might have noticed that the decreasing sales are caused by lower prices of the competitors. Therefore, the manager starts a search for new suppliers delivering products of a specific type (step 5). The screenshot in figure 4 shows the query interface with a query for suppliers delivering trousers with a price lower than 80 Euros. This query can be composed interactively by the user by browsing the ontology in a tree-like structure, and selecting relevant items for the query. The query interface is intelligent as it

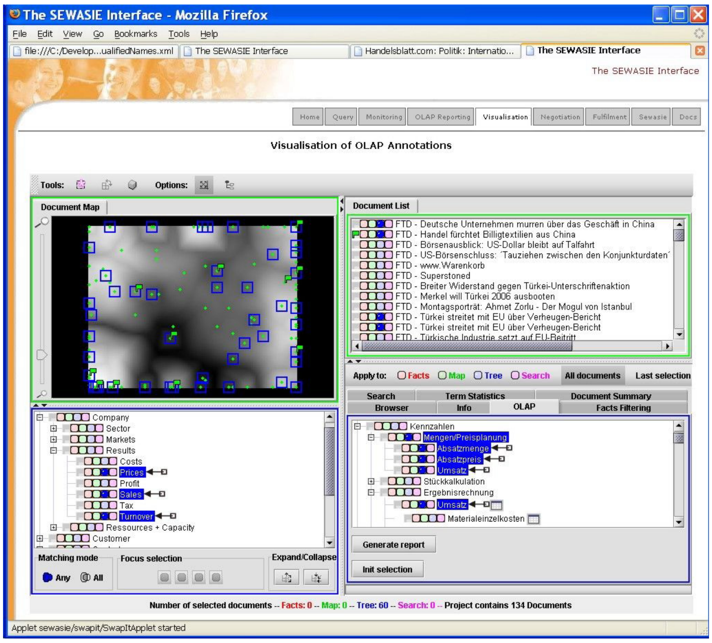

Figure 3: Monitoring & Visualisation Interface

contains an online reasoning functionality, i.e. it allows only a combination of items which is consistent wrt. the ontology.

If the query returns a list of suppliers, the user can select a subset of them and start a negotiation (step 6). The negotiation is integrated into the system in several ways. Firstly, the information of the query (e.g. products and their constraints) are transferred to the negotiation tool and are integrated into the first version of the business contract. Secondly, the negotiation is structured and ontology-based, i.e. the exchange of message follows a specific protocol and the contents of the messages is semantically enriched by relating it to concepts of an ontology (left part of figure 5). Finally, the content of contract is further formalised by the possibility of adding rules to the contract. These rules represent contractual conditions which have to be applied if certain exceptions occur (e.g. reduced price in case

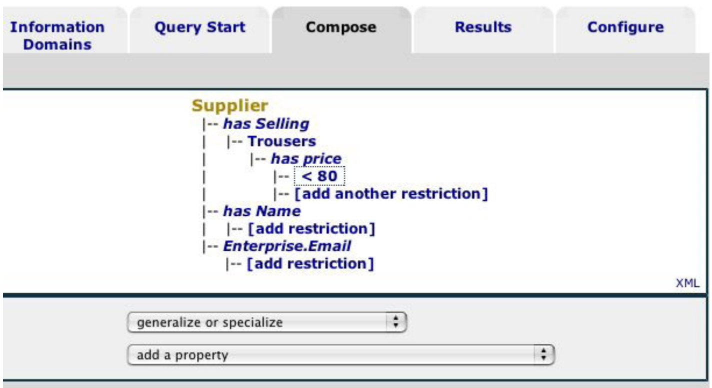

Figure 4: The query interface

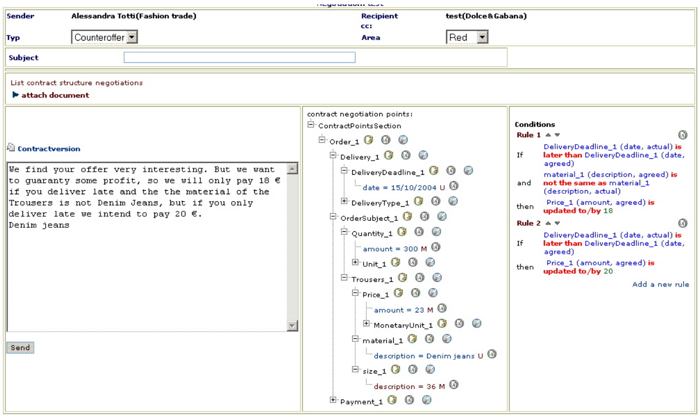

Figure 5: Ontology-Based Negotiation

of late delivery).

4.2 Operational Perspective

In the background, this scenario is supported by several intelligent agents. The queries are processed by query agents, which take into account the BA ontology provided the brokering agent. Note that there are two different types of mappings: the first mapping is done within a SINode and maps the data sources to the GVV of the SINode. The second type of mapping is at the brokering agent level and maps several GVVs of SINodes to one ontology of an

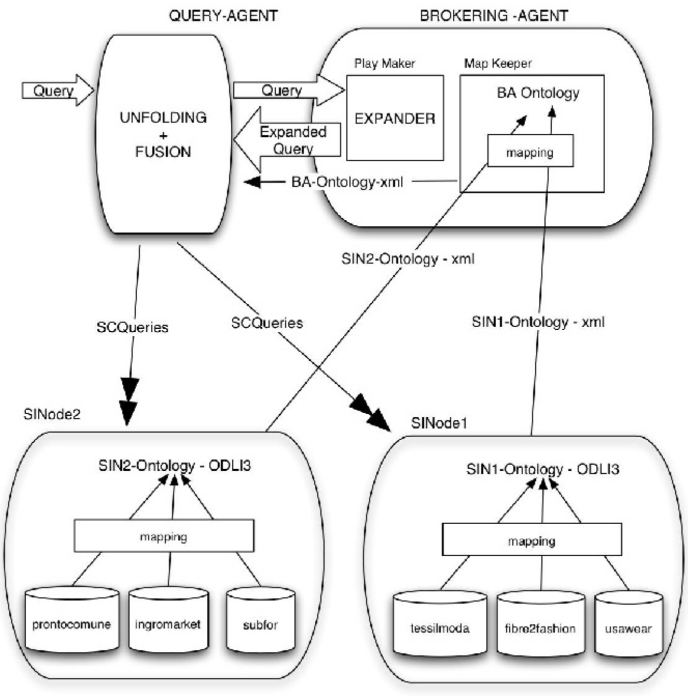

Figure 6: Query Processing in SEWASIE

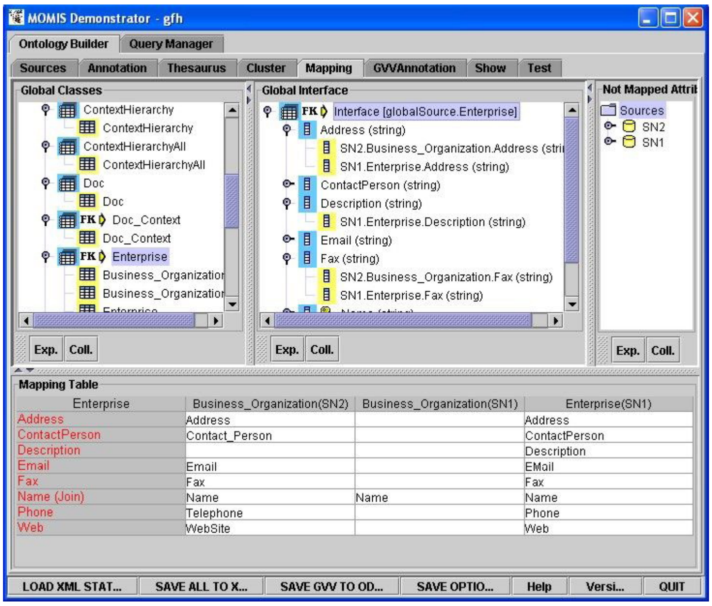

Figure 7: Mapping definition in the Ontology Builder

brokering agent. In general, the mapping from the data sources to the BA ontology is not simply the composition of the two mappings. We have shown that (in the case that the mappings are defined as global-as-view mappings) the mappings can be indeed composed, and the reformulation of a query to the BA ontology can be done in two steps (figure 6): first, the query is expanded and unfolded on the brokering agent level, and then a similar process is done within a SINode.

Queries are normally formulated by the user using the query interface as described before. The monitoring agent generates also queries based on the monitoring profile of a user or the annotation of an OLAP report.

4.3 Design Perspective

The various mappings between the data sources and ontologies have to be defined at design time by an administrator or ontology designer. The SEWASIE system provides several tools to support this task.

First, the data sources have to be mapped to the GVV of a SINode. This is done by using the Ontology Builder (OB) mapper which supports the tasks of mapping and integrating data sources in a semi-automatic way. Similarities between the schemas of the sources are detected automatically. The system generates “global classes” and relates these classes to “local classes” of the sources. We use the language ODLI3 (Object Definition Language with extensions for information integration) to formalise the GVV and the mappings to the data sources. The mapping which has been created by the system automatically can be fine tuned by an ontology designer using the graphical interface shown in figure 7.

A similar process is repeated at the brokering agent level. A first version of the BA ontology is bootstrapped using the same techniques as within a SINode. However, the BA ontology needs much more careful design as integrates several ontologies from different SINodes. Therefore, the SEWASIE system provides an ontology design tool (figure 8). The

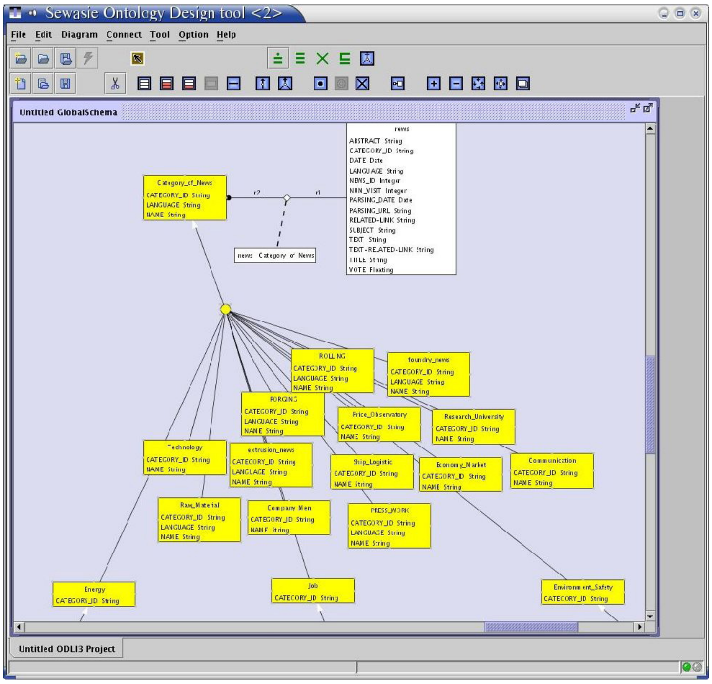

Figure 8: Ontology Design Tool

tool provides the usual feature of ontology editors (adding/removing of concepts and relationships). In addition, it is connected to a reasoner which enables consistency checks or the deduction of implicit relationships.