Metadata with a MISSION: Using Metadata to Query Distributed Statistical Meta-information Systems

Sally McClean, Bryan Scotney
School of Computing and Information Engineering, University of Ulster, Northern Ireland

Hans Rutjes, Jannes Hartkamp
Desan Research Solutions, The Netherlands

Abstract

This paper describes the technologies developed for the extraction and publication of large volumes of statistical information, such as are produced by public data collection organisations and National Statistical Institutes (NSIs); the approach has been implemented in the MISSION system that has been developed as part of a EU Fifth Framework project. We review the MISSION system, and then focus on our novel approach that utilises metadata to simplify the user interface and database query process, by allowing the user to browse metadata to compose a query by specifying the format of the answer. The system then decomposes the request into query fragments, works out what type of data could be used to answer these query fragments, and searches the metadata to find where such data can be found and what processing is required. Once the query has been composed, data and metadata are retrieved from the distributed sites, processed, combined appropriately and the result returned to the user.

Keywords: *Metadata-guided processing, Statistical meta-information systems.*

1. Introduction

The Web has had a profound impact on the way National Statistical Institutes and other data providers publish data. The world is moving towards a global market and providers of official statistics need to supply data in this environment. MISSION aims to provide a software solution that will address the issues raised by this context. It utilises the advances in statistical techniques for data harmonisation, the emergence of agent technology, the availability of standards for exchanging metadata and the power of Internet information retrieval tools. The result is a modular software suite aimed at enabling providers of official statistics to publish data and metadata in a unified framework, allowing users to share methodologies for comparative analysis and data harmonisation. The software modules are distributed over the web and communicate via agents.

The MISSION system is operational, and has been evaluated during its development via a workshop of external users, whose recommendations have been considered for advancing both theoretical issues and system implementation. A public MISSION server has been set up, from which interested parties can, first of all, download and install a Client that is preconfigured to connect to a remote Dataserver and Library, so that they can query the information in the system. The full system as it has been installed and configured can also be downloaded, including all sample data and mappings.

The MISSION system has a three-tier architecture that comprises three basic logical, or conceptual, units or building blocks, which can be deployed in different scenarios. The components are: The Client, The Library, and the Data Server. The Client is a down-loadable module connecting a user to a home Library. The Library is a repository for statistical metadata, holding different kinds of metadata to support searching, access and explanation. The user formulates his/her queries using a graphical interface supplied by the Client; such queries may be composed with the assistance of a browser, which utilises agents to liaise with the metadata repositories, which reside in the Library. The user interfaces via a query interface that specifies the context of a query (the frame). Figure 1 gives a general overview of how the MISSION system might be deployed.

The user first browses the metadata in order to compose a suitable query. The Client then sends the request to the Library, which analyses and decomposes it, sending queries to other Libraries if necessary. A series of query agents then analyse the query, and, based on searching the metadata in the Library, develop a plan for obtaining an answer. This involves decomposing the request into sub-queries, matching the query components to metadata describing the available datasets, and sending requests for the results of these sub-queries to the appropriate Data Servers, registered with the Library. Once the results of the sub-queries are known, the data and metadata are combined and the distributed data sources merged. The result is displayed by the Client in a graphical interface.

Datasets are incorporated into the MISSION system using the MISSION Importer. This allows micro and macro data to be imported into the Data Servers, and stored in relational databases. In addition the Importer sends metadata to the Library where it is stored and used by the

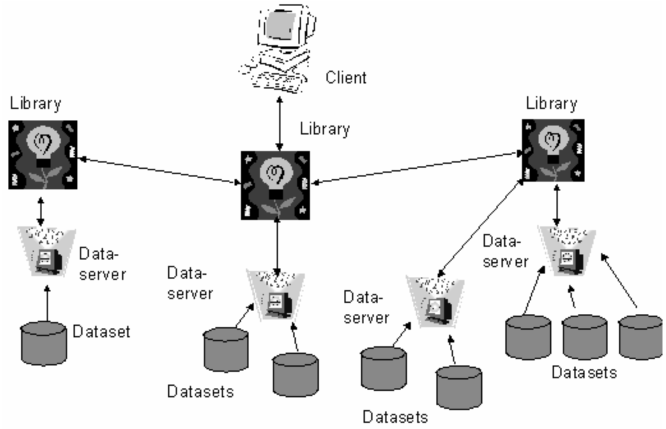


Figure 1. Deployment of the MISSION system

browser when constructing a query, and also used by the query agents when answering the query; such metadata holds information on the datasets that have been imported into the system. In order to use the Importer, the General User MISSION Interface (GUMI) Data Importer interface is employed. In addition, when a query accesses data on a Data Server, a Data Retriever Server is utilised to partially process queries before returning the result to the requesting Library.

In addition, to data import to the Data Servers, metadata may be managed, created and imported to the Classification Server in the Library using the Library Administration tool.

2. The Metadata

Within the MISSION system metadata is utilised for three activities:

- searching for data
- analysing data
- interpreting data.

The Library is a repository for the statistical metadata within the system. It holds three different kinds of metadata. The most basic type of metadata is *access* metadata which is the physical and logical information required to search for and access statistical data. The second kind is *methodological* metadata which is the information required to process that data in order to satisfy requests for statistical analysis. The third kind of metadata is *contextual* metadata that supplies background information and explanatory notes for the user, thus facilitating the interpretation of data. This kind of information includes, for example, the purpose of a survey, or an explanation of a break in series for a time series. This information can be attached as footnotes to a query result. The first two types of metadata are machine understandable. The last is machine readable and human understandable.

In the vocabulary of the Dublin Core Metadata Initiative (DCMI), the MISSION system is both software and a service, and its principal resources are datasets. The metadata model used by the MISSION system can be mapped onto the Dublin Core Metadata Element Set. The system has three types of users: Library Administrators, Data Providers, and Client Users. Both Library Administrators and Data Providers input metadata into the system. The Library Administrator defines frames. A frame is a means of grouping and constraining ontologies and their mappings to a shared geographical/ temporal area. Frames group ontologies that share the same statistical units and have some conceptual unity. Ontologies that are added to a frame must adhere to the specified geographical and temporal descriptors and apply to the specified statistical unit. In terms of the Dublin Core elements, a frame is defined using a title, a description, an identifier, a subject, and coverage (i.e., the geographical and temporal frame constants); the Library Administrator is usually the creator, contributor, and publisher.

A Data Provider inputs datasets into the MISSION system. These may be of a variety of types, including SPSS pre-processed datasets and PCAXIS datasets, and comprise both raw data and metadata. *Access* and related metatdata is provided via the GUMI interface; this includes identifier, title, server location, provider name, database type, retriever ID and IP, and classification server location, as well as some formatting information. Again, dataset constants are defined, corresponding to the geographical and temporal frame constants that describe coverage. For each dataset, access rights are set by the Data Provider for registered users. Notes, in the form of text in structured notes tables, are also an important part of the MISSION system. In MISSION, notes may be attached at various levels, e.g., cell, variable value label, variable, table, frame, and are treated as integral types of metadata that are concurrently processed by the system when statistical processing is carried out; ist is for this reason that the notes are modelled within a system of conceptual note table structures.

An important aim of the MISSION system is to provide user-friendly access to multiple data and metadata sources that are differently represented in terms of language and levels of detail but nonetheless represent semantically equivalent information. Typically a user might want to compare data sources from different countries where the variables may be differently named and one country may have collected data at a finer level of detail than the other. However, by mapping variable names and values onto each other and aggregating appropriately to harmonise the data sets, comparisons are possible.

In the language of the database community we say that such systems are heterogeneous in that there is a semantic mismatch between the database schemas (the values and variables); we will here use the term *ontology* to refer to a set of variables along with their values. Thus a crucial issue arising here is how to provide a uniform access mechanism for querying such databases. To support various aspects of users' activities, it not only requires a complex system architecture, but also necessitates a flexible data model for representing the content and location of statistical databases, and a smart user interface which provides users with the ability to browse ontology information, formulate complex queries, harmonize inconsistencies between ontologies, locate appropriate data, execute complex queries and output results with different layout.

Access metadata is extracted from the data sources when they are imported into the system. Such metadata contains information on the location, structure, access rights and ownership of the distributed data sources within the system. Also, when the data sources are imported into the system, contextual metadata, such as notes, are also input to the Library. Methodological metadata, in the form of descriptions of ontologies and mappings between ontologies, are stored in the Classification Server which also resides in the Library and contains information on the ontologies (that describe values and variables and associated values in the system) along with mappings between ontologies. These mappings allow the system to recognize and automatically harmonise data with semantically equivalent ontologies.

Preparing mappings between ontologies is a significant, and often complex, aspect of providing data, but is vital if data is to be useful in an automated heterogeneous environment. One of the challenges for the MISSION project has been to design mechanisms and procedures that enable both Data Providers and Librarians to readily input useful metadata into the system, both at the time of dataset registration, and subsequently, as required, for ontology mappings. Currently the system contains a limited set of frames that group ontologies into the areas of World Health Organisation data, Educational and School-leaver data, and Scandinavian population data, for which metadata has been provided by Data Providers using the GUMI Data Importer interface. Initially seeding the system is a laborious process. However, as the system expands, the burden of dataset registration diminishes with the increased opportunities for providing metadata via mappings to already existing metadata. For example, completely new ontologies are less likely to be required, as it becomes increasingly possible to use, or to slightly edit, already existing ones. Central to the system is the Classification Server, whose role in providing mappings between ontologies is greatly assisted by the existence of internationally recognised classifications that operate as "hubs" for providing mappings between ontologies in general.

Classifications are the foundations on which all statistical systems, national as well as international, are built. However, the existence of a variety of classifications as well as their revisions raises problems of compatibility and comparability of data collected and disseminated in a distributed environment. Having such independently developed classification schemes raises a number of requirements such as:

- The user might pose a query and require the results to use a specific classification and its nomenclature; we call this the *ontology*. This may be a local classification or an internationally recognised classification.
- The system requires to find matches to the sub-queries, by searching through metadata from each local data site stored in *Libraries* (metadata repositories). We here achieve this functionality via a matching agent.
- Once a set of matches is found, the query plan is constructed. This must include mappings generated by the negotiation agent so that the query fragments can be re-written in the local data site's ontology. In addition, the query result must be translated back into the ontology of the user.

Aspects of such problems have been discussed in the literature of other related fields, principally:

- Ontology research in computer science - both in the context of the Internet and also, increasingly, for distributed heterogeneous databases.
- Schema matching, primarily in the database field, where the mappings between heterogeneous schema are learned.
- Information brokers and Information integration over the Internet.
- Strategies that the statistical agencies have employed in dealing with heterogeneous classifications and nomenclatures.

Here we use the definitions of classification provided by the METANET group [2] and the Neuchatel group [6]. A classification is defined as a structured list of mutually exclusive categories, each of which describes a possible value of the classification variable. Such a structured list may be linearly or hierarchically structured.

In [1] an *ontology* is defined to be an explicit specification of a conceptualisation. In [8] an ontology is defined to be a shared understanding of some domain of interest. We use the term ontology to refer to a set of variables along with their classifications. Thus, an ontology might be a survey, e.g., UK Labour Force Survey 2001, whereas particular categorical variables within the ontology will have classifications. A set of ontologies, along with some mappings between them, we call a *frame,* e.g., UK Labour Force Survey 2001, UK Labour Force Survey 2000, UK Labour Force Survey 1999, is a frame.

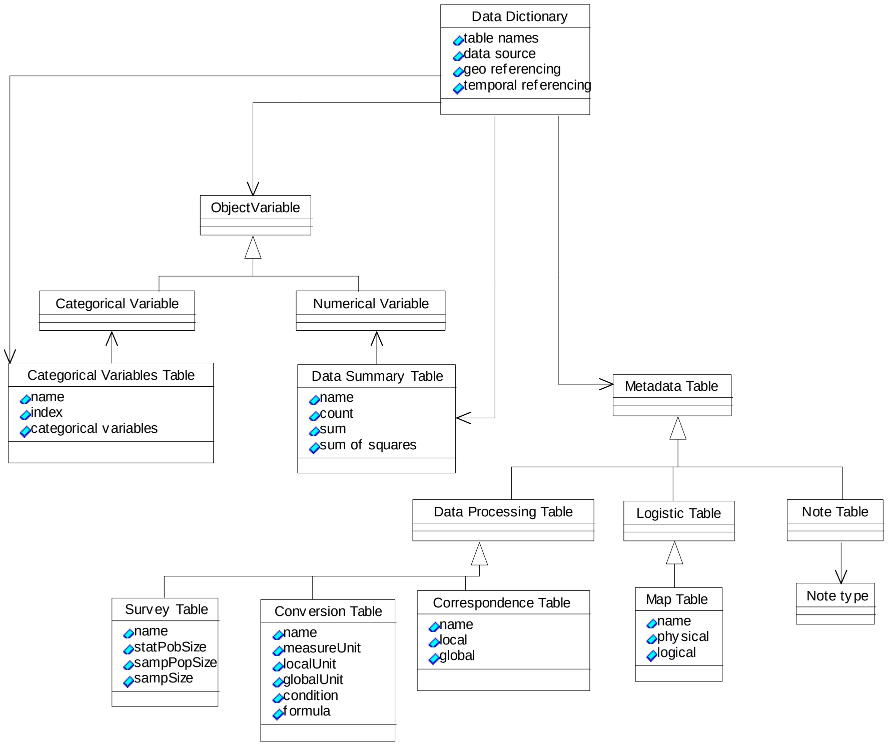


Figure 2. The Mameob

When the query result is computed from the various sub-query results, it is stored and processed in the form of a Macro-Meta-Data-Object (Mameob) where the data are stored between data summary tables and categorical variables tables. Associated metadata tables, such as the note tables, store the passive metadata. Figure 2 describes the various components of a Mameob.

3. Using the Metadata

3.1 Specifying the query

In this section we will concentrate on the *active metadata* within the system that is used to decompose the query, locate suitable data sources to answer the query fragments, construct the query execution strings, locate and retrieve the data and (*passive*) metadata, and unify the query fragments. We use the term active metadata to mean metadata involved in processing (the access and methodological metadata); in this context contextual metadata is passive.

The query is constructed in the Client via the Browser. The Browser View presents an ontology as a tree; this tree is extracted from the Classification Server as an XML file. The user can switch from a view with general ontology information (e.g., coutries) to another view with specific information (e.g., specific values). The query construction employs a "query by example" style as a means of query formulation, and incorporates the layout definition into the query formulation process. Thus the user describes the variables, values and layout of the query result; the system automatically works out how to achieve this result. When a user starts to formulate the query, he/she first navigates the ontology information, and then drags an individual node of the query tree such as "Ireland" or group nodes, such as "Gender" and "Type of School"; these are dropped onto the query view.

Such querying requires a basic query language and a basic query structure that can interface with the statistical operators and the data and metadata model introduced above. Such a basic query language and structure is shown in Table 1.

The queries are constructed within the Query Constructor; simultaneously the Query Editor automatically rewrites the query into the Table Query Language (TQL). The TQL describes the query in terms of a basic ontology (values and variables) and geo- and temporal- references. Hence, unlike standard database SQL, the TQL provides a declarative approach in which the query is specified in terms of the desired output, rather than in operational terms. This is an essential aspect of the system, since the data sources to be used are *a priori* unknown to the user. The query is then decomposed by the query agents, on the basis of the geo- and temporal- references; these represent different datasets in the native data. The different clauses of the TQL are translated by the query agents into different types of processing, requiring different statistical operators

Operator	Operand	Example of operand
COMPUTE	Table	Table, graph or model
OF	n	n, mean or s.d.
ON	Context	Survey, e.g. LFS
FOR	Target concept	Numerical attribute e.g. salary
BROKEN-DOWN-BY	Cross-product of categorical attributes	GENDER by JOB
WITH	Predicate	GENDER = Female
OVER	Geo-referenced categorical attribute(s)	EU-Countries
IN	Temporal-referenced categorical attribute(s)	YEAR = 1990 thru 1999
ONTOLOGY	Ontology	WHO classification as defined in …

Table 1. The Table Query Language (TQL)

(Figure 3). When the query has been specified in the Query Constructor, a description of the query is then passed to the Library for processing.

The table query objects in the TQL directly relate to statistical operators that operate on the Mameobs. The result of a statistical operator on a Mameob is also a Mameob. Thus this part of the system constitutes an ***algebra.*** The relationship between the TQL and the Mameob is presented in Figure 4.

Use cases for such a system involving agents interacting with basic Mameobs and operators are shown in Figure 5.

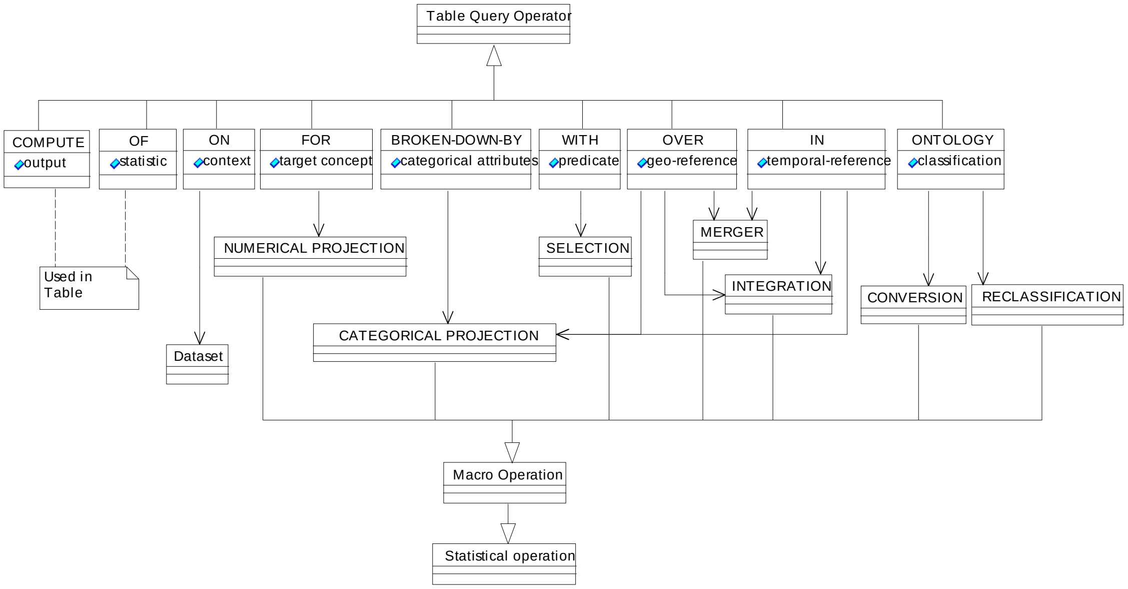


Figure 3. The relationship between the TQL and the statistical operators.

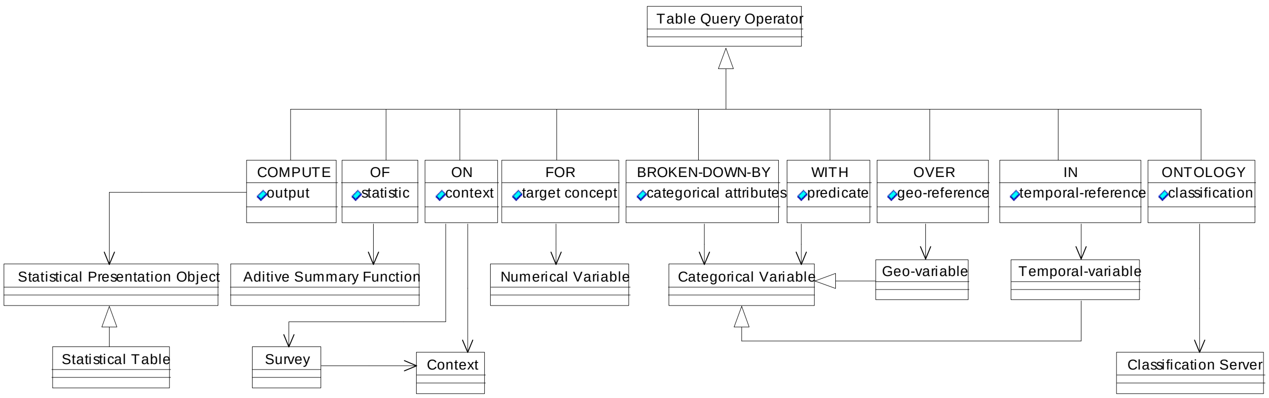


Figure 4. Relationship between the TQL and the Mameob

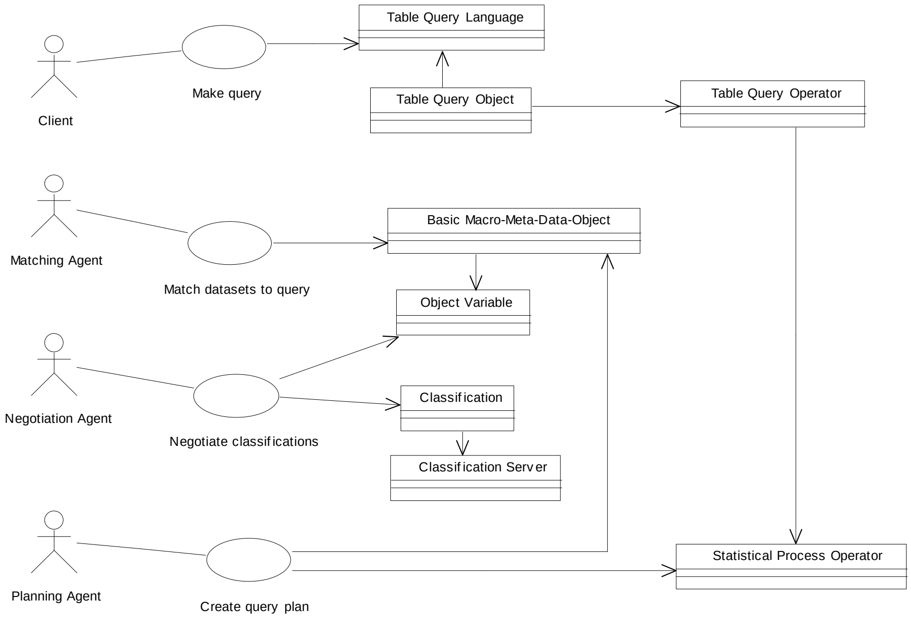


Figure 5. Use cases for various agents

3.2 The Agent Architecture

In MISSION heterogeneity is resolved and harmonisation achieved via the negotiation agent, which is called in by the matching agent in consulatation with the Classification Server, to decide how the sub-query is best covered by the candidate datasets [3]. The covering agent then composes the best combination of sub-queries and the planning agent converts the optimal cover into a plan (defined in terms of statistical operators) and manages the execution of that plan. If only a partial match is made, the matching agent may use a negotiation agent to determine if a full integration is possible. The negotiation agent is utilised to determine if different classification schemes can be mapped onto each other via a common ontology. This task is carried out using classification servers. These tasks have been termed *pre-integration*.

Other Query Agents then construct an operator stack to transform the data to match the (sub) query. The distributed databases are accessed via brokering agents in liaison with the costing agents and information agents. Here brokering agents have the capability of learning other retrieval strategies if there is a problem with the optimal strategy (as constructed by the covering agent). Costing agents (including authentication) are responsible for determining costs of retrieving various data fragments from possible data sources; possible costs are monetary, Internet transportation time, and processing time. Information agents act on behalf of the data sources. These tasks comprise the *integration* and are carried out once the integration strategy has been formulated by the pre-integration process.

- The query is communicated to the system via interface agents. Interface agents keep user profiles at the client site allowing the system to construct queries in an intelligent way tailored to the user's characteristics.

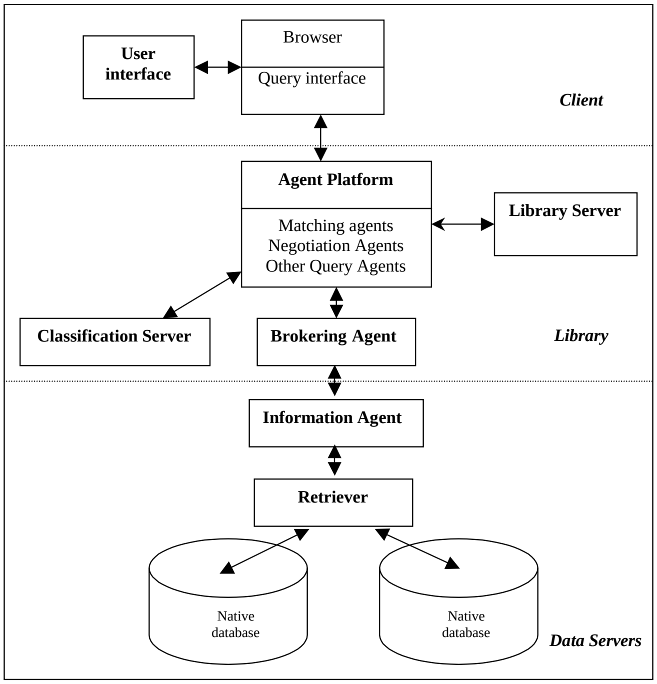


Figure 6. The internal architecture of the MISSION system

Such a system provides a flexible and elegant way of providing access to heterogeneous distributed statistical databases, allowing the common ontology to be constructed on-the-fly once a query has been specified. Further details are provided in [4].

The matching agents decompose the query into sub-queries, and then search metadata in the Library Server to find datasets that match the query fragments. The internal architecture is presented in Figure 6.

Previous work [7] has developed a global data model that enables a universal, harmonised analysis of the data and the metadata. Specific mappings, procedures, functions and algorithms have been developed to enable transformations from local to global views of the data; this permits local-as-view processing, along with query re-writing at run-time to transform to global-as-view. However, this approach requires the data providers to map their data to a global ontology that may be quite laborious. Our current approach, on the other hand, has a number of advantages over previous research, namely:

- The data providers can choose to map their data to (different) classifications, available on the Internet. This is a less laborious solution.
- The query may be posed in a local ontology defined by the user – a query-as-view solution. There is therefore no global ontology, as such. Instead, the ontology mappings and query re-writing rules are computed dynamically. This is clearly more flexible than previous approaches.
- By employing a Negotiation Agent, the task of constructing the query plan is substantially automated compared with previous approaches.

4. A "Walkthrough": The Mission System in Operation

4.1. The Data and the Ontologies

We consider an example using data from the countries of Finland, Norway, and Sweden using a Frame *Nordic*. The datasets within the *Nordic* frame have the following variables:

- Age
- Citizenship
- Sex
- Time

There are two ontologies within the *Nordic* Frame:

- *Nordic-fine*
- *Nordic-coarse*

In both ontologies, the variables citizenship, sex, and time have the same classification. The values for these three variables are:

Citizenship:	*Africa, Asia, Canada, Denmark, Finland, Iceland, Norway, Oceania, Stateless, Sweden, USA, Unknown, other America, other EEA, other Europe*
Sex:	*Male, Female*
Time:	*1999, 2000, 2001*

The variable age is classified differently in the *Nordic-coarse* ontology from the *Nordic-fine* ontology.

In the *Nordic-fine* ontology age is classified using the five-year groupings:
0-4 years, 5-9 years, 10-14 years, 15-19 years, 20-24 years, 25-29 years, 30-34 years, 35-39 years, 40-44 years, 45-49 years, 50-54 years, 55-59 years, 60-64 years, 65-69 years, 70+ years.

In the *Nordic-coarse* ontology there is a coarser classification:
0-4 years, 5-9 years, 10-14 years, 15-19 years, 20-29 years, 30-39 years, 40-44 years, 45-49 years, 50-54 years, 55-59 years, 60-64 years, 65-69 years, 70+ years.

Ontology mappings have been made that map the *Nordic coarse* age classification to the *Nordic-fine* age classification, as shown in Figure 7.

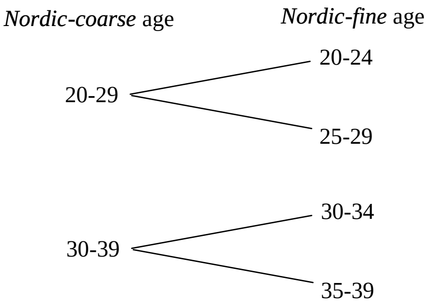


Figure 7. Ontology mappings for age classification

All other mappings between variables and values are made automatically, based on the exact matching of variable names and value labels.

4.2. The Query

Consider the query defined by the Query Table in Figure 8. Leaving the WITH field blank means that all values of the variables Sex and Age in the *Nordic-coarse* ontology will be used in the query (i.e., the default SELECTION is “all values for all variables in BROKEN-DOWN-BY”).

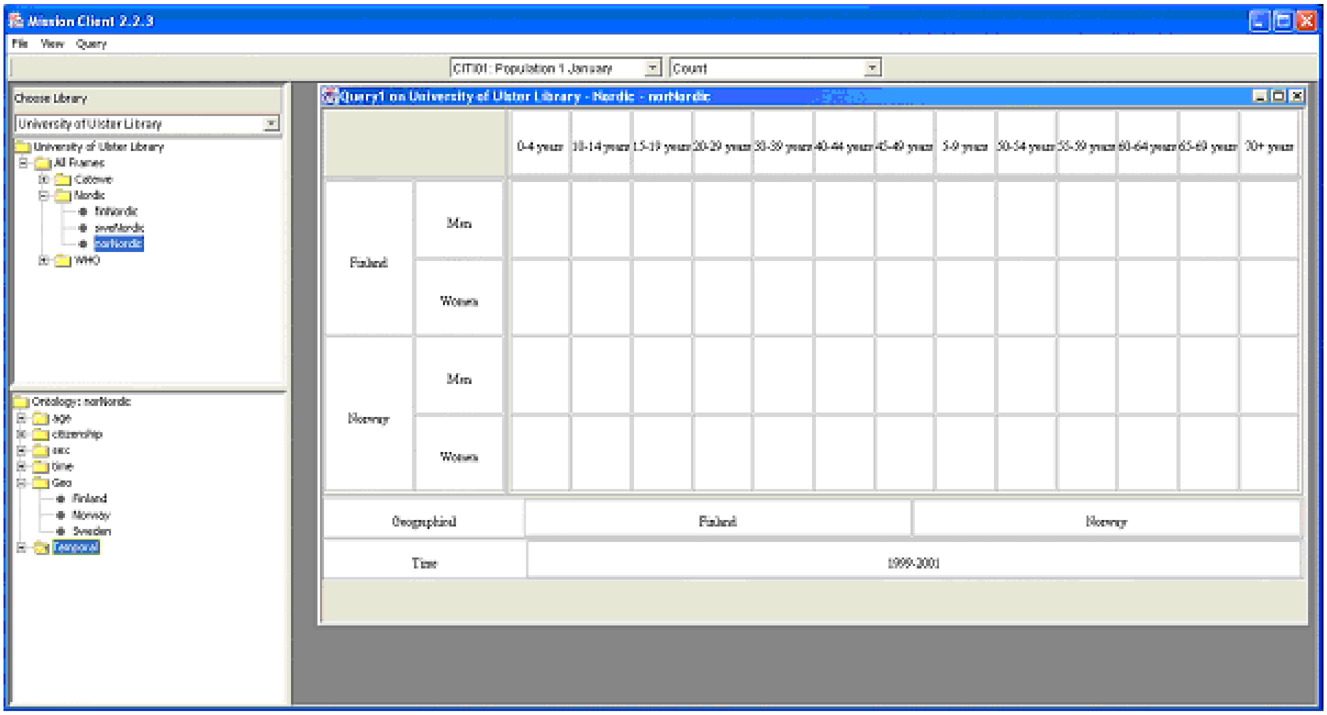
A SHALLOW MERGE over the two geographical values, Finland and Norway, is required, i.e., we wish to see the query answered separately for Finland and Norway and the results placed in the same table. This is achieved by placing the two Geo values, Finland and Norway, into the stub of the Query Constructor, as shown in Figure 9.

We note that a DEEP MERGE could be achieved by omitting the Geo values from the Query Constructor stub. In a DEEP MERGE, the query results for Finland and Norway would be “pooled”.

Once the query has been specified using the Query Constructor, it may be viewed (and amended) using the Query Editor, as shown in Figure 10.

Operator	Example of operand
COMPUTE	Table
OF	counts
ON	*Nordic*
FOR	CITI01: Population 1 January
BROKEN-DOWN-BY	Sex by Age
WITH	
OVER	Finland, Norway
IN	1999-2001
ONTOLOGY	*Nordic-coarse*

Figure 8. An example of a Query Table



CITI01: Population 1 January

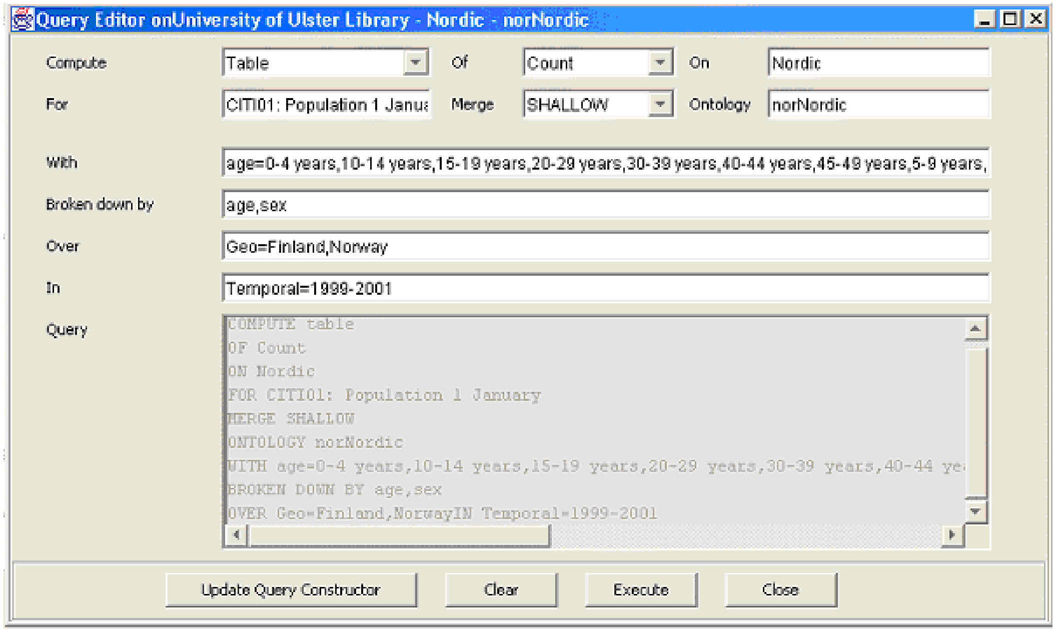
Count

Query1 on University of Ulster Library - Nordic - norNordic

		0-4 years	10-14 years	15-19 years	20-29 years	30-39 years	40-44 years	45-49 years	5-9 years	50-54 years	55-59 years	60-64 years	65-69 years	70+ years
Finland	Men													
	Women													
Norway	Men													
	Women													

Geographical	Finland	Norway
Time	1999-2001	

Figure 9. The Query Constructor


Query Editor onUniversity of Ulster Library - Nordic - norNordic

Compute	Table	Of	Count	On	Nordic
For	CITI01: Population 1 Janua	Merge	SHALLOW	Ontology	norNordic

With: age=0-4 years,10-14 years,15-19 years,20-29 years,30-39 years,40-44 years,45-49 years,5-9 years,

Broken down by: age,sex

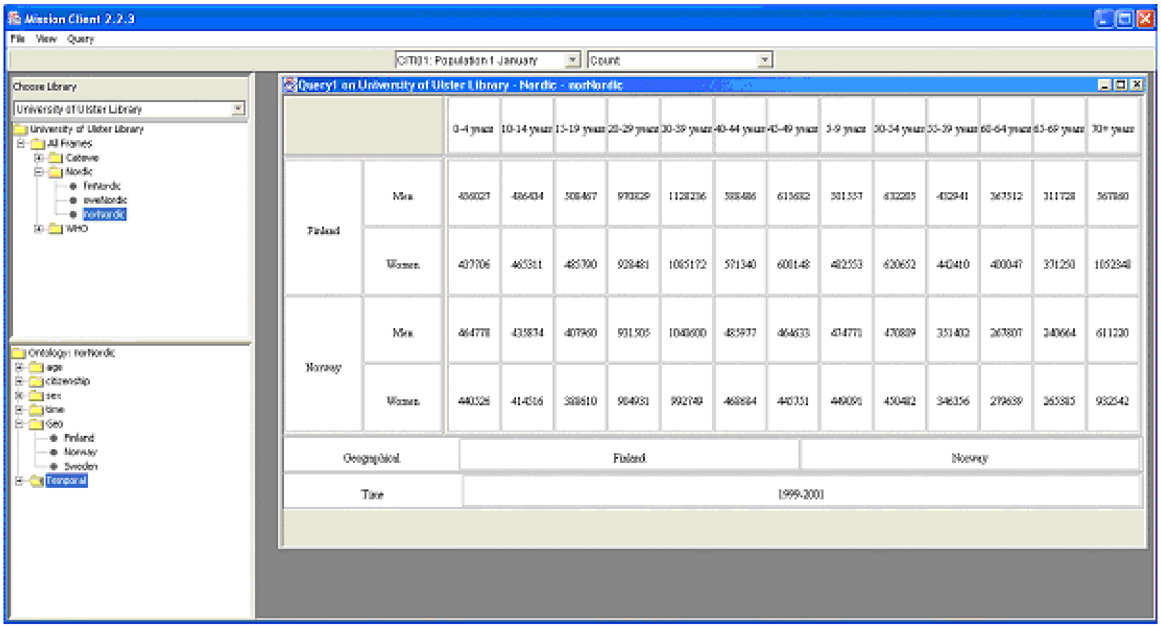
Over: Geo=Finland,Norway

In: Temporal=1999-2001

Query:

```
COMPUTE table
OF Count
ON Nordic
FOR CITI01: Population 1 January
MERGE SHALLOW
ONTOLOGY norNordic
WITH age=0-4 years,10-14 years,15-19 years,20-29 years,30-39 years,40-44 ye
BROKEN DOWN BY age,sex
OVER Geo=Finland,NorwayIN Temporal=1999-2001
```

Figure 10. The Query Editor



Query1 on University of Ulster Library - Nordic - norNordic

		0-4 years	10-14 years	15-19 years	20-29 years	30-39 years	40-44 years	45-49 years	5-9 years	50-54 years	55-59 years	60-64 years	65-69 years	70+ years
Finland	Men	456027	486404	508467	978829	1128236	588486	615682	501537	632205	432941	367512	311728	567860
	Women	437706	465311	485790	928481	1085172	571340	600148	482553	620652	440410	400047	371250	1052948
Norway	Men	464778	435874	407960	931505	1040600	485977	464633	474771	470809	351402	267807	240664	611220
	Women	440526	414516	388610	904931	992749	468684	440751	449091	450482	346356	279639	265385	932542

Geographical	Finland	Norway
Time	1999-2001	

Figure 11. Query result in Query Constructor interface

4.3 Query Execution

When the query is executed, the following sequence of actions occurs. First the Client specifies the query. Then a Matching Agent locates datasets that have the potential to answer the query. For this purpose, the Geo values, Finland and Norway, and the Temporal value, 1999-2001, are matched against metadata held in the Library. A Covering Agent checks that the query can be covered by the query fragments available from the potentially useful datasets. The cost of retrieving the data for each query fragment from each of the potentially useful datasets is found using a Costing Agent. In case more than one potential match is found for a given query fragment, these costings are used to prioritise the datasets to be used. The Importer uses the Database Retriever to retrieve the required data from the provider datasets that have been matched to the query fragments.

Statistical processing operators are then used to generate the results that answer the query from the datasets that have been identified as useful. This processing may include reclassification if a dataset is being used for which the ontology is different from the ontology of the query. To achieve this, ontology mappings held in the Classification Server are used. Statistical processing of the retrieved datasets takes place in the Library Platform, and the result that answers the query is returned to the Client for display in the Query Constructor interface. The query result is shown in Figure 11.

5. Summary and Further Work

MISSION encountered a number of theoretical questions during the construction of the system, as we tried to move from the ideas to a practical solution. We have not solved all of them satisfactorily, but have tried to find a real world balance between serving the needs of the 'casual' end user and the analysis specialist. This is an ongoing problem found in all applications making statistical data available over the web. In particular we had to decide whether to maintain our own vocabulary (ontology and frame) or to use more 'user oriented' terms. We finally accepted the former solution, as the concepts involved, e.g., 'data dictionary', 'code list', were different from the more commonly used ones.

The latest version of the system has solved crucial firewall issues that would have hampered installation and testing by interested parties. Although firewalls have little to do with the software as such, and their widespread use and exact functioning could not have been foreseen at the start of the project, MISSION has acknowledged that their existence is a crucial fact, especially in organisations in the public domain, but increasingly in other organisations as well. For dissemination purposes, the software has been adapted in such a way that it can function irrespective of firewall settings: the full system can be installed, queries can be specified and results returned. The only module whose functioning depends on firewall settings is the Client Workspace, which allows end users to specify, save and use their own ontologies and mappings.

Exploitation of the MISSION system will consist of consultancy, implementation of tailor-made solutions and maintenance of the systems. Separate from the MISSION system, the Agent Platform has generic possibilities, and will be registered as open source software.

We have described the MISSION system, which provides an agent-based solution to the automated querying of a distributed statistical meta-information system.

Metadata are used to compose the query, in a goal-based manner that describes the layout and content of the result. Agents then use additional metadata to identify, locate and process statistical data and metadata, combined and processed in the form of a macro-meta data object.

Such an approach provides a capability of automating the process of executing queries on heterogeneous statistical databases that permits queries to be specified in a goal-driven query-by-example format. Rather than impose an a priori global standard, the user can query through a unified interface, and integration is done at run-time. Further work will extend this aspect of the query process to allow for an inexperienced user to make an imprecise query, without specifying an ontology. The system then automatically constructs a dynamic shared ontology by analysing the correspondence graphs that relates the heterogeneous classification schemes [5].

Such ideas relate to developments in the Semantic Web. A key challenge for the Semantic Web is to discover new knowledge from distributed databases that are semantically heterogeneous. Such semantic differences may come about when databases have evolved separately, with post hoc semantic mappings between schema (ontologies) later being constructed. However, with the advent of pervasive and ubiquitous computing, using small-scale and mobile devices, it is increasingly likely that semantic heterogeneity will originate from variations in scale. In further work we propose to develop a flexible method for Knowledge Discovery from semantically heterogeneous data, based on the specification of ontology mappings and the automation of data harmonization and processing, using appropriate metadata.

Acknowledgement

This work was funded by MISSION- Multi-agent Integration of Shared Statistical Information over the (inter)Net (IST project number 1999-10655) within EUROSTAT's EPROS initiative.

References

[1] Gruber, T. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5.

[2] METANET (2000). http://www.epros.ed.ac.uk/metanet/

[3] McClean, S.I., Páircéir, R., Scotney, B.W., & Greer, K.R.C. (2002). A Negotiation Agent for Distributed Heterogeneous Statistical Databases. *Proc. 14th IEEE International Conference on Scientific and Statistical Database Management (SSDBM)*, 207-216.

[4] McClean, S., Karali, I., Scotney, B., Greer, K., Kapos, G.-D., Páircéir, R., Hong, J., Bell, D., & Hatzopoulos, M. (2002). Agents for Querying Distributed Statistical Databases over the Internet. *International Journal on Artificial Intelligence Tools*, *11*(1): 63-94.

[5] McClean, S.I., Scotney, B.W., & Greer, K.R.C. (2003). A Scalable Approach to Integrating Heterogeneous Aggregate Views of Distributed Databases. *IEEE Transactions on Knowledge and Data Engineering*, *15*(1): 232-236.

[6] Neuchatel paper, Version 2 (2000). Available from claude.macchi@bfs.admin.ch

[7] Papageorgiou, H., Pentaris, F., Theodorou, E., Vardaki, M., & Petrakos, M. (2003). A Statistical Metadata Model for Simultaneous Manipulation of both Data and Metadata. *International Journal of Intelligent Systems*, forthcoming.

[8] Uschold, M., & Grüninger, M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, *11*(2), 1996.