UNIVERSITY OF WESTMINSTER

WestminsterResearch
http://www.wmin.ac.uk/westminsterresearch

Context based querying of scientific data: changing querying paradigms?

Epaminondas Kapetanios[*]
Moira Norrie
Swiss Federal Institute of Technology, Department of Computer Science, Institute of Information Systems, Switzerland

Bernard Brabec
Swiss Federal Institute for Snow and Avalanche Research, Switzerland

[*] Epaminondas Kapetanios is now employed by the Harrow School of Computer Science, University of Westminster.

Copyright © [1999] IEEE. Reprinted from 3rd IEEE Metadata Conference, 06-07 April 1999, Maryland, USA.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Westminster's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

The WestminsterResearch online digital archive at the University of Westminster aims to make the research output of the University available to a wider audience. Copyright and Moral Rights remain with the authors and/or copyright owners. Users are permitted to download and/or print one copy for non-commercial private study or research. Further distribution and any use of material from within this archive for profit-making enterprises or for commercial gain is strictly forbidden.

Whilst further distribution of specific materials from within this archive is forbidden, you may freely distribute the URL of WestminsterResearch. (http://www.wmin.ac.uk/westminsterresearch).

In case of abuse or copyright appearing without permission e-mail wattsn@wmin.ac.uk.

Context-based Querying of Scientific Data: Changing Querying Paradigms?

Epaminondas Kapetanios and Moira C. Norrie

Swiss Federal Institute of Technology (ETH)
Institute for Information Systems
ETH-Zentrum, CH-8092 Zurich, Switzerland
email:{kapetani, norrie}@inf.ethz.ch

Bernhard Brabec

Swiss Federal Institute for Snow and Avalanche Research
Flueelastrasse 11, CH-7260 Davos Dorf, Switzerland
email: brabec@slf.ch

Abstract

We are investigating and applying a semantically enhanced query answering machine for the needs of addressing semantically meaningful data and operations within a scientific information system. We illustrate a context based querying paradigm on the basis of a Regional Avalanche Information and Forecasting System - RAIFoS which is concerned with the collection and analysis of snow and weather related physical parameters in the Swiss Alps. The querying paradigm relies upon the issue of interactively constructing a semantically valid query rather than formulating one in a database specific query language and for a particular implementation model. In order to achieve this goal, the query answering machine has to make inferences concerning the properties and value domains, as well as data analysis operations, which are semantically valid within particular contexts. These inferences take place when the intended query is being constructed interactively on a Web-based blackboard. A graph-based display presentation formalism is used with elements including natural language terms, measurement units, statistical quantifiers and/or specific value domains. A meta-data database is used to organise and provide the elements of the graph each time the graph, and consequently the intended query, is expanded or further refined. Finally, the displayed graph is transformed into elements of the implementation model from which, in turn, SQL statements and/or sequences of statistical operations are created.

Keywords: *Scientific Information Systems, Meta-data, Query languages, Finite State Automata, Conceptual structures, Statistics, Semiotics*

1 Introduction

Posing queries to database systems presupposes a full understanding of a particular implementation model, for example, a relational schema, in terms of structures and data values. Moreover, one needs to learn the syntax of a database specific query language in order to formulate a syntactically correct query. In the case of scientific data systems, the situation is often exacerbated due to the large size of the database schemas and the extensive use of schema and value encodings. This, in turn, has a major impact on the query results which become a subject of investigation by data analysis operations. Addressing the semantically appropriate data for further processing such as data analysis and visualisation, presupposes an adequate interpretation of the underlying data schema(s) and values.

For this reason, it is necessary to provide an abstract description of the contents of a database, or databases, which enables users to construct meaningful queries without requiring knowledge about the implementation model. The term `meta-data` has been used extensively in the literature to refer to metacircular descriptions of implementation models, for example, tables describing tables in relational databases, classes or collections describing other classes or collections in object-oriented ones, etc. On the other hand, the term *meta-data* has been closely related to the terms `knowledge base` [12, 14, 3, 4] and `ontologies` [25, 24, 9, 7, 2, 15] in describing application domains, especially when more than one data source is concerned.

However, even when posing queries at a meta-data level [10, 8, 16, 23, 26, 1], there si typically no support for the interpretation of values and/or data analysis operations. Moreover, the user still needs to be familiar with the underlying query language syntax. Database query languages, in general, have not focussed on the integration of syntax and semantics. It turns out that semantically incorrect queries may be executed and this, in turn, can have a major impact on overall system performance, especially when large amounts of data are involved.

On the other hand, a navigational style of traversing classification hierarchies standing for descriptions of database contents does not provide the full expressiveness of a query language based on a set manipulation algebra. This approach has a rather descriptive character but cannot be used as a query language itself, where operations and/or complex properties are considered.

Therefore, we departed from the conventional querying paradigms and moved into a context based querying paradigm which relies on a semantically enhanced querying answering machine. It resembles a finite state automaton, the interpreter of which, helps in interactively constructing an intended query in terms of a semantic network. Query construction is considered as a movement through semantically consistent information states by activating particular nodes of a graph as presented on a Web-based blackboard. Nodes are the terms of the query language and links the underlying syntax. The terms are natural language based interpretations of constructs of a database schema, such as relations, classes and both one- and multi-dimensional attributes, as well as measurement units, statistical quantifiers and context based specific values domains.

The intended query is expressed in terms of a graph based formalism and, therefore, it is easy to be made equivalent to, for example, a relational calculus based formalism. Since the interpreter uses the interpretation of the underlying database schema and data values in terms of meta-data to infer next states, only syntactically correct queries are considered which are also semantically meaningful for data analysis purposes.

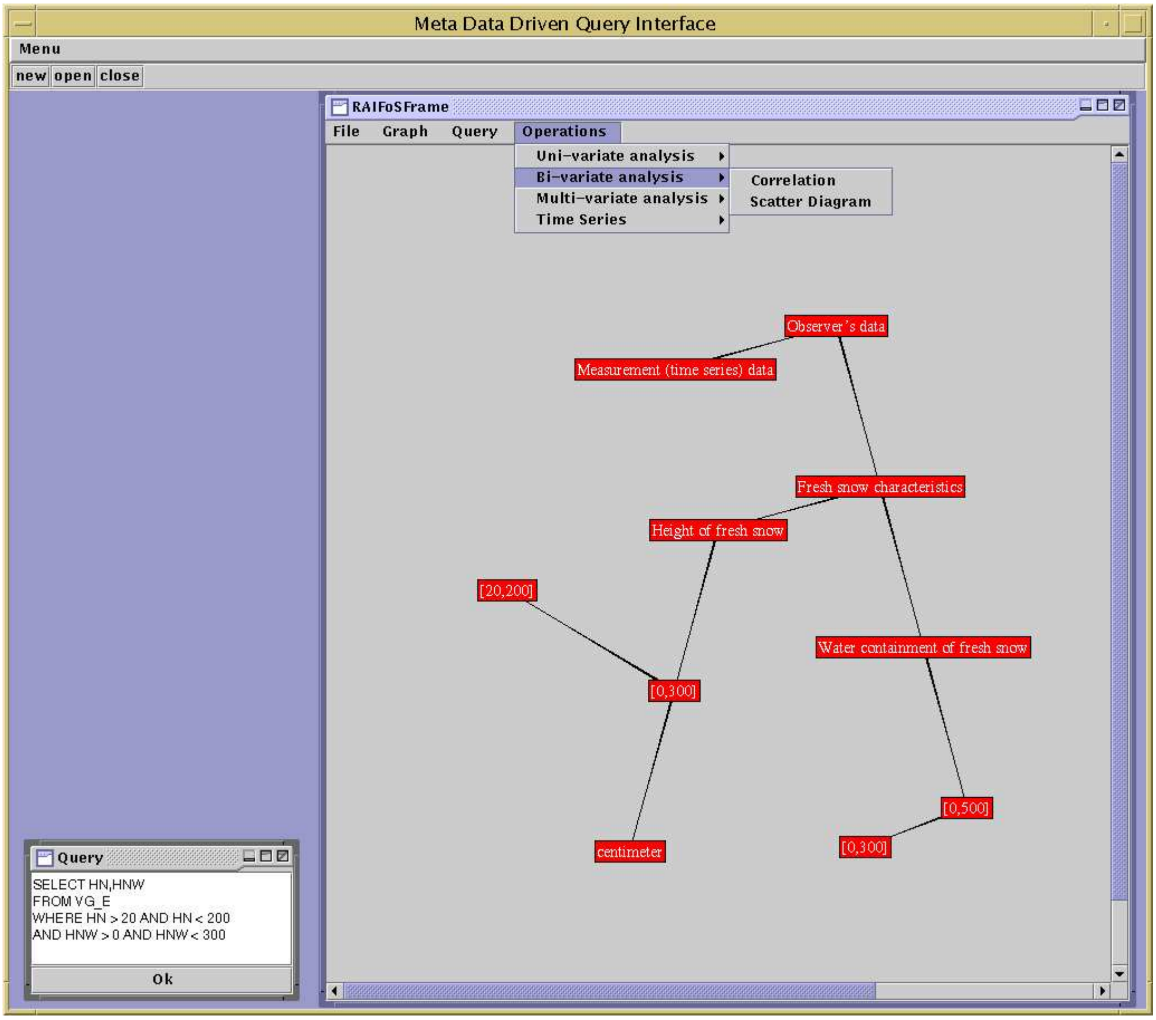
2 Background

The **R** egional **A**valanche **I**nformation and **Fo**recasting **S**ystem - RAIFoS - is being developed to meet the requirements of querying snow and weather related measurement data as collected by various measurement stations in the Swiss Alps. Measurement stations belong to different measurement networks and refer to geo-locations. Furthermore, 50-60 different physical parameters or variables are being measured which are assigned particular measurement units. All data are stored in operational databases built upon relational engines and/or file systems. For storage efficiency purposes, most of the non-numerical data have been encoded, for example, the regions of Switzerland are assigned numerical codes.

In order to analyse and/or visualise data, scientific users are currently required to pose queries in terms of a data model, schema and value encodings which may be unfamiliar. Moreover, they need to learn the syntax of a database specific query language, usually SQL. Thus, it turns out that the requirements for operating on data in a semantically correct way involve

- learning the query language syntax
- understanding the underlying data schema in terms of interpretations of relations, classes, collections, associations holding among them, and attributes
- interpretation of value encodings as well as measurement units in which domain values are expressed

Research goal: In order to cope with these requirements, we elaborated a semantically enriched query answering machine for addressing scientific data. Instead of trying to formulate a query in a particular database query language with interpretations based on assumptions, the query answering machine should be able to help in



```
SELECT HN,HNW
FROM VG_E
WHERE HN > 20 AND HN < 200
AND HNW > 0 AND HNW < 300
```

Figure 1: A snapshot of the metadata-driven query interface

constructing a semantically valid query. This should be done by a querying interface such that the construction of queries becomes a matter of navigation through information states. In other words, addressing a particular set of data values to be analysed is enabled through interaction with a meta-data server which represents knowledge about the data schema and values in terms of mappings between data model constructs and value encodings, and elements of the querying interface as presented to the end-user.

Figure 1 depicts a snapshot of the elaborated querying interface. It relies on a graph-based representation formalism. The end-user can start with an initial keyword or concept to be considered for the intended query. This is represented by a node on the blackboard. Links to related concepts and/or properties can be built up dynamically when a particular node is activated by clicking on that node. Nodes are mostly labelled with natural language terms. Further elements to appear on the blackboard as part of the constructed graph are `specific value domains` and `measurement units` to be considered in conjunction with particular properties.

Importantly, the system should infer the new nodes which are semantically associated with the activated ones in terms of a given context. In other words, an inference mechanism should check out the current context as presented in terms of nodes and links on the blackboard and infer the new state into which one can move.

For example, `Measurement (time series) data` is considered as a central concept to be addressed. By activating this node, links will be established to all semantically associated concepts and/or properties which can be queried in conjunction with this concept. `Measurement stations` is such a concept indicating the fact that measurement data has been delivered by particular measurement stations. Further links can be established to categories of measurement data such as `observer's data`. By activating this node, links are drawn only to those properties (physical parameters) which are expected to be measured by observers. This, in turn, leads to nodes standing for specific value domains, for example,`[0-300]` for the height of fresh snow. Furthermore, they must be expressed in specific measurement units, for example, `cm`. Note that links to particular properties and/or specific value domains vary according to a given context which for physical experiments is mostly given by space-time structures.

When the constructed graph is in a final state, the user can activate the query generation process, which transforms the selected graph into the database specific query language, or can trigger a data analysis operation. For example, 1 depicts a screenshot of the metadata-driven query interface with a generated SQL query. Note that particular value intervals can be specified as input, the range of which should ly between the boundaries as given by $[0-300]cm$ value interval. It is also possible to address more than one properties in the same graph.

3 Research methodology

In order to achieve the goal of addressing data and operations in a meaningful way, we elaborated and partially implemented a `Semantically Enhanced Query Answering Machine` (SEQAM), the main components of which are depicted in figure 2.

Information States Blackboard: As illustrated in figure 1, the information states blackboard is the querying interface of the system. It consists of two parts: one dealing with the graph construction and the other dealing with the results presentation - we will return to this part in the last paragraph of this section. Since we are interested in providing such a querying interface over the Internet, a Java based implementation of these components has been realised using the AWT library as provided by JDK-1.1.x. However, the querying interface can also be used as a stand-alone application.

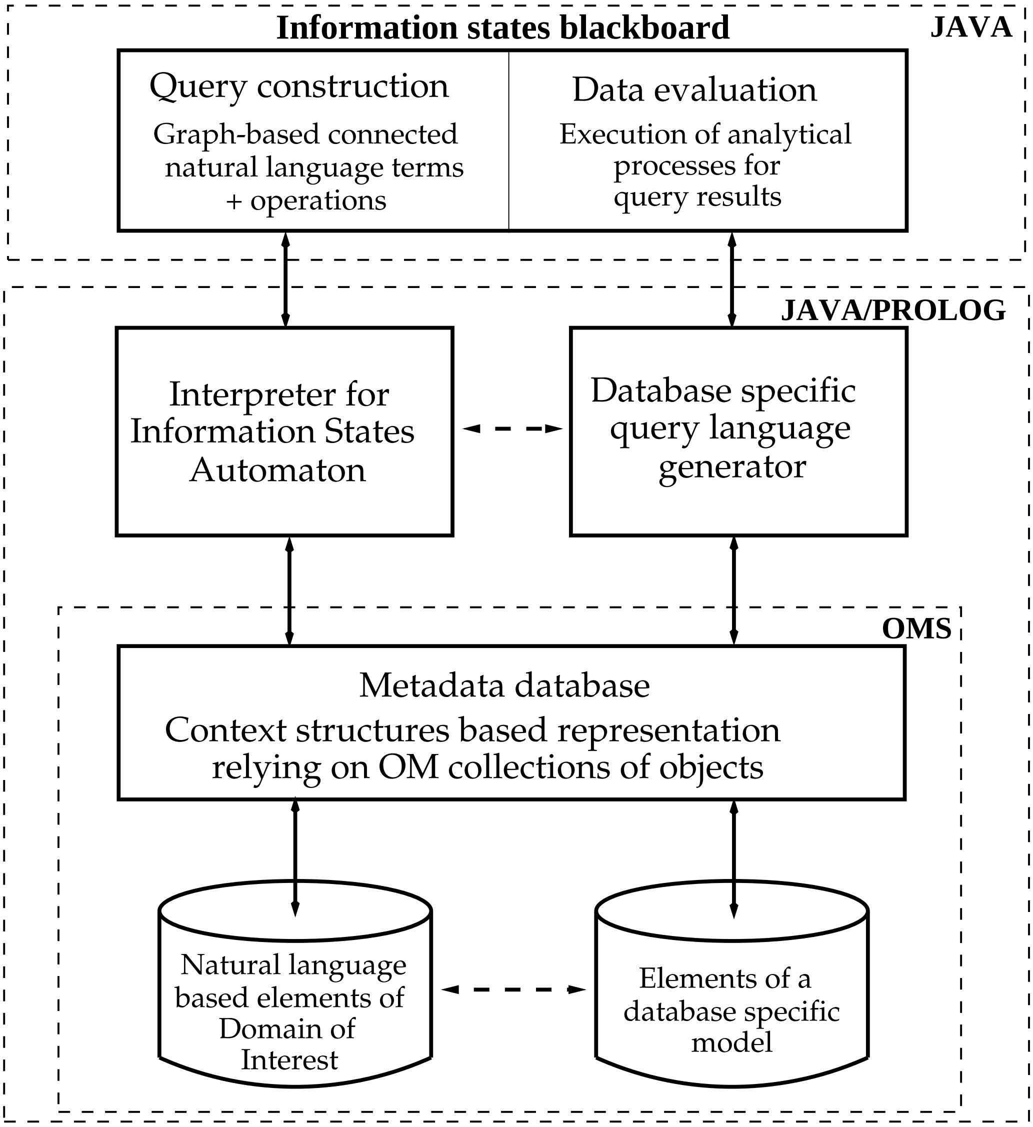


Figure 2: Architecture of the semantically enhanced query answering machine

As a starting point, an initial word expressing the central element, over which a query is going to be constructed, is selected or typed in and presented as an initial node on the blackboard. Subsequently, activating nodes by clicking on them will cause all semantically linked elements to be displayed as neighbouring nodes. Through this interaction, the user is guided to the construction of a query rather than trying to formulate it directly. This happens with respect to *syntax and semantics* as related to an underlying implementation model. The final state resembles a query expressed by nodes and links of a graph, where nodes stand for the elements of an input alphabet and links for the syntax of the grammar. The interactive construction of the graph relies on an interpreter which is formally presented in the next paragraph.

Access to the remaining components of SEQAM takes place either over specific OMS database communication protocols such as JDBC-OMS [6] and Internet-OMS [5], or by using the native API of Prolog to be called from Java. The latter is considered as the most effective and elegant solution, since the implementation of the interpreter for the information states automaton and the generator of database specific queries are strongly oriented towards symbol manipulation and, therefore, Prolog is the most elegant platform for their implementation.

Interpreter for Information States Automaton: Since query construction becomes a matter of moving through information states, we rely on an interpreter for an information states automaton, the specification of which resembles those of non-deterministic finite automata (NDFA). Finite automata, as a mathematical model of a system with discrete inputs and outputs, describe the system in terms of a finite number of internal configurations or "states". Each state summarises the information concerning past inputs which are needed in order to determine the behaviour of the system on subsequent inputs. In computer science, the theory of finite automata has been used for the design of text editors or lexical analyzers as finite state systems. The central issue is the control of acceptance of subsets of symbols belonging to a given alphabet [17, 18, 19, 11].

In particular, the NDFA for information states is defined as a 5-tuple $(Q, \Sigma, \delta, I, F)$, where

1. Q is a finite set of states,
2. Σ is a finite input alphabet,
3. δ is the mapping from $Q \times \Sigma$ to Q,
4. I is the initial state,
5. $F \subseteq Q$, is the set of final states.

If δ is a function, then the finite automaton is deterministic. Σ, the input alphabet, is specified as the collection of meta-data language terms, a subset of which might constitute an intended query. The elements of the alphabet are equivalent to the nodes of the graph as presented on the information states blackboard.

The alphabet Σ of meta-data terms is mainly categorized into: a) the subset of natural language terms standing for, not only the description of data model constructs such as relations, classes, relationships or attributes, but also the data values corresponding to value encodings; b) measurement units; c) statistical quantifiers; d) specific value domains such as particular interval or categorical values which might describe arithmetic intervals, for example, the values `low`, `high`, `medium` etc. describing particular arithmetic intervals for a given parameter.

Each time a node is activated on the information states blackboard, a new element of Σ is actually selected. The interpreter checks out the new information state into which the user could move by applying the mapping δ. The definition of the mapping δ is such that only semantically meaningful new states are considered within a given context. Therefore, the interpreter is always aware of the given context.

For example, a particular property might not come into question for a particular context, or, a specific value domain should be addressed only within a specific context. In RAIFoS, this enables us to deal with semantic dependencies such as the fact that only specific Alp regions should be considered in an intended query, if a particular network of measurement stations is being addressed. This, in turn, has a major impact on the restrictions to be taken into account within the conditional part of a query statement and may lead to significantly focussed results.

The initial state I is a meta-data term - an initial node on the blackboard - which might also be considered as the entry point for further investigation of information states. Hence, we try to `construct a semantically coherent query with the help of the interpreter rather than formulating a query`.

The final state F is actually the intended query to be posed.

Meta-data Database: The purpose of the `meta-data database` is twofold:

a) To provide a representation platform for structuring and managing the meta-data language terms as elements of the input alphabet Σ in such a way that new states can be inferred, i.e. the mapping δ can be applied. Additionally, semantically meaningful operations can also be inferred. We will refer to this part of the meta-data database as `domain of interest`.

b) To provide a representation platform for mappings holding between Σ and elements of the implementation model(s) for the *database specific language generator* (see also figure 2). We refer to this part of the meta-data database as *implementation symbols*. Both collections of elements will be referred to together as `meta-data terms`.

In order to achieve these goals, the representation platform should rely on an object-oriented paradigm with database management facilities for the structure and manipulation of meta-data terms. In particular, the representation technique should enable the elements of Σ to be structured in such a way that contexts can be expressed as well as associations holding among the elements of Σ. The latter indicate the syntax of the potential meta-data queries, i.e. the form of the constructed graph. Hence, the grammar of the meta-data language is captured by `binary associations` holding among the elements of Σ.

It turns out that inferences of new states are based on, not only the syntax of the alphabet, but also on the semantics as expressed by context structures which introduce a `depth` into the syntax. In order to express contexts, object identifiers are crucial. The same term of Σ might be associated to different neighbouring terms due to the given `situation` or `context`. For example, the term `temperature` is associated to the terms for the value domains [-50..50] and [-50..10] depending on the given context, namely, association to the terms `air` and `snow`, respectively.

Introducing depth into the syntax means that rich classification structures must also be available. It helps in coping with further semantic issues which are crucial for the operation inferences and/or the generation of database specific languages. Operation inferences rely on the fact that particular operations addressing one- or multi-dimensional properties make sense only in consideration of their classification due to the value contents. For example, an `average function` makes sense only if it refers to a property the values of which, although numerical, are not interpreted as encodings of the gender. Similarly, plotting operations for data analysis purposes depend on the interpretation of the value contents as *categorical* or *numerical* [13], as expressed by the corresponding classification of the property(ies).

Further classification structures of the *domain of interest* indicate the issue of transforming the constructed graph into a database specific language in terms of the *implementation symbols*. It has been proven that graph-based formalisms for languages are equivalent to other formalisms such as logic-based or λ-calculus, upon which most conventional query languages rely. However, finding the adequate SQL-statement, for example, requires going through the mappings between elements of *domain of interest* and *implementation symbols* as well as inferences made on the basis of their classifications. Therefore, elements of the *domain of interest* and, consequently, elements of the input alphabet Σ are mainly classified as `Concepts`, `Properties` or `Specific Value Domains`, in order to provide a most general term for *relations*, *classes*, *attributes*, *fields* and *value types*. This is particularly useful when more than one implementation model is considered.

It turns out that the syntax of the semantically enhanced meta-data language should be expressed by a `multi-layered directed cyclic graph`. This formalism underlies the inferences made by the interpreter of the information states automaton as well as the generator of database specific query languages (see figure 2).

However, manipulation of *meta-data terms* is cumbersome without an adequate representation and management mechanism. Such manipulations may arise when elements of the input alphabet Σ are restructured by changing contexts and/or "depth", or enhancing of the meta-data alphabet. Further, elements for implementation symbols must be reconstructed when changing or adding implementation models.

For this purpose, we rely on OMS as an object-oriented database system to provide all constructs and functionality required for representing and managing multi-layered graph-based formalisms. The system relies on the OM model which addresses the issues of *collections of objects* and *directed binary associations* as well as *classification structures for both objects and binary associations* [22, 21, 20].

Database specific language generator: Assuming that measurement data reside on relational sites, the constructed graph has to be transformed into an SQL equivalent statement. At the current stage of development, AND-connected SQL statements, possibly with natural joins, are generated. We claim that the generated SQL statement is semantically meaningful in terms of the further processing of the delivered results since only semantically coherent graphs and operations are allowed by the interpreter of the information states automaton.

The transformation algorithm relies mainly on the classification of the elements of the input alphabet Σ as `concepts`, `properties` and `specific value domains`, together with their mappings to the elements of the set of `implementation symbols`, in order to fill in the SELECT-FROM-WHERE pattern. However, further considerations of the implementation model are taken into account concerning physical storage optimization aspects as well as meta-data tables referring to the underlying model specifications such as primary and foreign keys, attribute types, etc.

The following example concerns the generation of the SQL query

```
SELECT THS0, THS1, THS2, THS3, THS4
from ENET1H_E
where stao_abk = "DVS"
```

from the constructed graph depicted in figure 3.

The linear representation of this graph corresponds to the following syntax:

$snow \rightarrow sensored\ temperatures$
$measurement\ network \rightarrow automatic\ measurement\ station$
$automatic\ measurement\ station \rightarrow 1 - hour\ frequency$
$snow \rightarrow Alps\ region$
$Alps\ region \rightarrow Davos\ region$
$Davos\ region \rightarrow Davos$

where `snow`, `sensored temperatures`, `automatic measurement network`, `Davos`, etc. are all elements of the input alphabet Σ. These elements are accepted by the interpreter for the information states automaton when starting with the initial state consisting of the element `snow`.

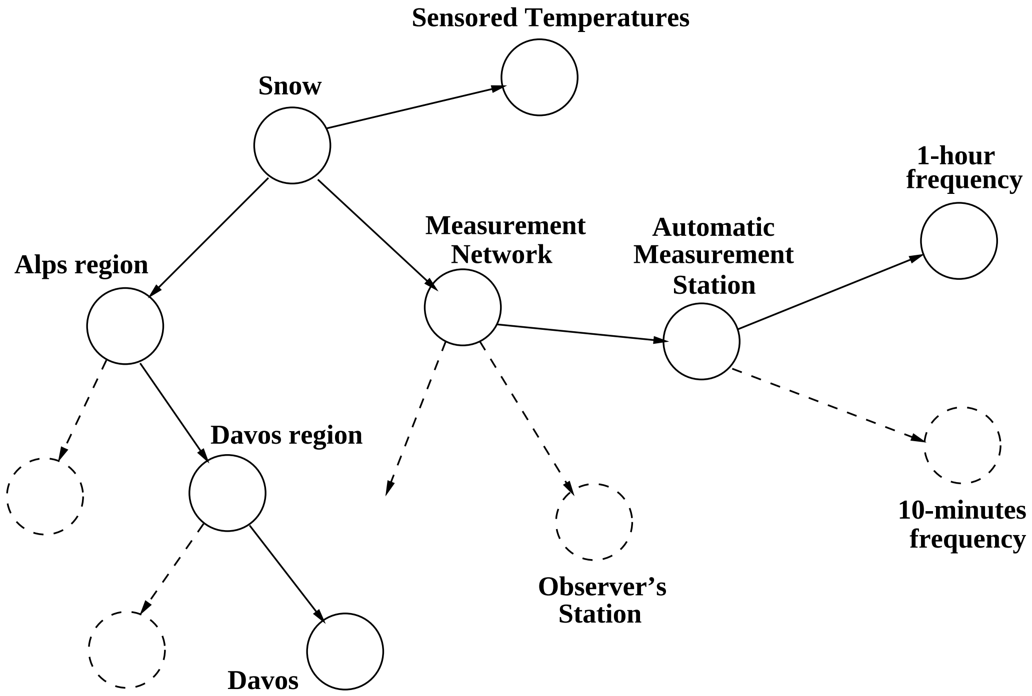


Figure 3: Graph based representation of the intended query

Note that attributes THS0, THS1, THS2, THS3, THS4 refer to temperature data measured at various snow levels above ground surface. They are automatically placed within the SELECT-FROM-WHERE pattern, since the selected element of the meta-data language is `sensored temperatures`, a term which covers all of these cases. Similarly, we can make use of more abstract terms which refer to multi-dimensional properties including space-time structures.

The transformation algorithm becomes more complicated when more than one implementation model must be considered. This happens to be the case when more than one data source is addressed. Since this issue is not the focus of this paper, we will not refer to these problems in detail.

Data evaluation: We refer to this part of SEQAM as the component responsible for the execution of the selected data analysis operation in conjunction with the intended query as expressed in terms of the constructed graph on the blackboard. These operations might be any kind of data analysis and/or plotting functions which contribute to the understanding of behaviour of data. They are provided by data analysis software packages such as $S-Plus^{TM1}$ and/or customized analysis algorithms. The results of these operations should appear on the information states blackboard.

---

[1] S-Plus is a trademark of MathSoft Inc.

Due to the seamless interface between query construction elements and data analysis operations, it was possible to hide all implementation details concerning the statistical operations and functions. These operations can be triggered directly from the *Information States Blackboard*, whereby the properties of the constructed graph, as they refer to data residing on an Oracle database platform, are considered as input to the statistical functions.

At a first stage of development, we mainly encapsulate functions referring to *descriptive statistics*, where one, two or three parameters can be taken into account. Correspondingly, we address *uni-variate*, *bi-variate*, *multi-variate* and *time-series* plotting functions such as `pie, bar, dot charts and histograms`, as well as `2/3-d plotting functions`. Summaries such as `mean and median values`, or `quantiles` are also considered.

4 Conclusion

We provide a querying paradigm for scientific data by adding a semantic layer to the underlying data-holding components which enables the system to guide users to data of interest, instead of requiring them to formulate database specific queries explicitly. Guidance takes place in terms of semantic structures which support the correct interpretation of encodings and/or data contents. Additionally, semantic constraints are taken into consideration such that context-based construction of information states is enabled. Data analysis operations can be triggered in conjunction with the constructed graph (intended query) from the information states blackboard.

A meta-data database for the management of the query construction elements has been implemented on the basis of the object-oriented database system OMS which is an appropriate realisation platform for a multi-layered graph formalism. However, as a first stage, the system is limited to the generation of AND-connected SQL statements. For the future, we will address the issues of generating SQL-queries covering all aspects of the relational calculus. Moreover, we intend to also cover the aspects of query generation for multiple (heterogeneous) data sources and/or languages.

References

[1] R. Agrawal, A. Gupta, and S. Sarawagi. Modeling Multidimensional Databases. Research Report CA 95120, IBM Almaden Research Center, 650 Harry Road, San Jose, 1995.

[2] Y. Arens, C. Y. Chee, C.-N. Hsu, and C. Knoblock. Retrieving and Integrating Data from Multiple Information Sources. *Int. Journal of Intelligent and Cooperative Information Systems*, 2(2):127–158, 1993.

[3] Amit Basu and Rafiul Ahad. Using a Relational Database to Support Explanation in a Knowledge–Based System. *IEEE Transactions on Knowledge and Data Engineering*, 4(6):572–581, December 1992.

[4] C. Collet, M. Huhns, and W.-M. Shen. Resource integration using a large knowledge base in carnot. *IEEE Computer*, pages 55–62, December 1991.

[5] A. Erni, M. C. Norrie, and A. Kobler. Generic Agent Framework for Internet Information Systems. In *Proc. IFIP WG 8.1 Conference on Information Systems in the WWW Environment*, Beijing, China, July 1998.

[6] T. Füstös. OMS Internet Integration Using Standard Interfaces. Master's thesis, Department of Computer Science, ETH Zurich, 1998.

[7] C. A. Goble, P.J. Crowther, and W. D. Solomon. A Medical Terminology Server. In *Proc. of the 5th Intern. Conf. on Database and Expert System Applications (DEXA 94)*, Lecture Notes on Computer Science, pages 661–670. Springer Verlag, 1994.

[8] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data Cube: A relational aggregation operator generalising group-by, cross-tab, and sub-totals. In *Proceedings of the 12th Inter. Conf. on Data Engineering ICDE'96*, pages 152–159, New Orleans, February 1996.

[9] N. Guarino and P. Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N.J.I. Mars, editor, *Proc. of the 2nd Intern. Conf on Building and Sharing Very Large Knowledge Bases*. IOS Press, Enschede, The Netherlands, 1995.

[10] Marc Gyssens and Laks V. S. Lakshmanan. A Foundation for Multi-Dimensional Databases. In *Proc. of the 22nd VLDB Conference*, Bombay, India, 1996.

[11] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Company Inc., 1979.

[12] T. Imielinski and H. Mannila. A Database Perspective on Knowledge Discovery. *Communications of the ACM*, 39(11):58–64, November 1996.

[13] R. A. Johnson and G. K. Bhattacharyya. *Statistics: Principles and Methods.* John Wiley and Sons, Inc., 3rd edition, 1996.

[14] Hatonen K., Klemettinen M., Mannila H., and Ronkainen P. Knowledge discovery from telecommunication network alarm databases. In *Proceedings of the 12th Inter. Conf. on Data Engineering ICDE '96*, pages 115–122, New Orleans, February 1996.

[15] D. Lenat and R.V. Guha. *Building Large Knowledge–Based Systems: Representation and Inference in the Cyc Project.* Addison–Wesley, Reading, MA, 1990.

[16] Shen W. M., Ong K., Mitbander B., and Zaniolo C. *Advances in Knowledge Discovery and Data Mining*, chapter Metaqueries for Data Mining, pages 375–398. AAAI, 1996.

[17] G. H. Mealy. A method for synthesising sequential circuits. *Bell Systems Technical Journal*, 34(5):1045–1079, 1955.

[18] E. F. Moore. Gedanken experiments on sequential machines. *Automata Studies*, pages 129–153, 1956.

[19] E. F. Moore. *Sequential Machines: Selected Papers.* Addison-Wesley, Reading, Mass., 1964.

[20] M. C. Norrie. An Extended Entity-Relationship Approach to Data Management in Object-Oriented Systems. In *12th Intl. Conf. on Entity-Relationship Approach*, pages 390–401, Dallas, Texas, December 1993. Springer-Verlag, LNCS 823.

[21] M. C. Norrie. Distinguishing Typing and Classification in Object Data Models. In *Information Modelling and Knowledge Bases*, volume VI, chapter 25. IOS, 1995. (originally appeared in Proc. European-Japanese Seminar on Information and Knowledge Modelling, Stockholm, Sweden, June 1994).

[22] M. C. Norrie, A. Steiner, A. Würgler, and M. Wunderli. A Model for Classification Structures with Evolution Control. In *Proc. of the 15th Int. Conf. on Conceptual Modelling (ER'96)*, Cottbus, Germany, October 1996.

[23] Meo R., Psaila G., and Ceri S. A new SQL-like operator for mining association rules. In *Proceedings of VLDB96*, 1996.

[24] Borgo S., Guarino N., and Masolo C. Stratified Ontologies: the Case of Physical Objects. In *European Conf. on AI (ECAI96)*, Budapest, Hungary, August 1996. Workshop on Ontological Engineering.

[25] Borgo S., Guarino N., Masolo C., and Vetere G. Using a Large Linguistic Ontology for Internet-Based Retrieval of Object-Oriented Components. In *Proc. of 9th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE 97)*, Madrid, Spain, 1997.

[26] Imielinski T. and Virmani A. A Discovery Board Application Programming Interface and Query Language for Database Mining. In *Proceedings of KDD96*, pages 20–26, Portland, Oregon, August 1996. AAAI.