The Use of Categorical Pullbacks for Integrating Heterogeneous Systems

B.N.Rossiter, Computing Science, Newcastle University, Newcastle upon Tyne NE1 7RU, UK.
B.N.Rossiter@newcastle.ac.uk
Michael Heather, University of Northumbria at Newcastle NE1 8ST, UK.

Introduction

Data warehousing attempts to integrate fully all the information in a business. Data mining attempts to discover knowledge. These processes are particularly difficult when heterogeneous systems are involved or where imperfect engineering standards have been applied in the development of information systems. In this paper we examine the requirements for mining data warehouses and develop a formalism for providing a solution that is both more general and more powerful than existing methods.

In technical terms, the theoretical problems in mining data warehouses crystallize into an emphasis on powerobjects, the need for novel connections and rules to be identified between powerobjects, the ability to construct the universal relations both intra-schema (local universe) and inter-schema (global universe), and a resolution of type/domain conflicts.

Requirements for Data Warehouses

At its face value an exact theory for mining data warehouses would expect to satisfy three points:

- deep interrogation is needed to discover exact knowledge;
- discovery of some knowledge might not be the complete answer;
- development of automatic methods is needed for knowledge discovery.

However these need to hold at every level. There may be a number of relevant levels. The question at any level is what are the conditions at another level. They may be quite different. Consistent reliable knowledge discovery therefore needs not only to be systematic but also universal.

In technical terms, the theoretical problems on mining data warehouses crystallize as follows:

- there is an emphasis on powerobjects rather than atomic objects with flexible searching required on clusters and groups;
- novel connections and rules need to be identified between powerobjects regarded as clusters or groups of data;
- universal relations needs to be constructed to make new connections intra-schema (local universe) and inter-schema (global universe);

- type/domain resolution is necessary to recognize which attributes are joinable.

Levels and Universality

These requirements for mining data warehouses call for a universal method rather than connections on a local basis. Relational models, perhaps using SQL, are based on set theory which is satisfactory for specific connections between data but is inadequate for some of the exploratory requirements above. Methods in object-oriented systems effectively provide strongly-typed functions which lack the flexibility needed. Indeed both of the standards for these two main techniques were thought to be unsuitable as the basis for federated databases at a recent conference [Eaglestone & Masood 1997; Roantree & Murphy 1997]. The object-based OQL lacks views in the relational sense and the relational SQL-3 has an *ad hoc* approach to typing.

Set theory logic is adequate for database models but: is it adequate for database of databases (universal) methods? Higher-order logic is needed which can naturally deal with openness in distributed systems and this is found to be topos logic [McLarty 1995].

Any theory of data warehousing therefore needs a sophisticated multi-level capability because of this further issue of universality. When operating only at a single level, distinctions between local and global conditions seldom arise. A simple example is dealing with dates: if they are all English documents a date like 2/5/97 raises no problems, but in an international set of documents, the user cannot be certain whether it refers to the 2nd May or 5th February. The convention used operates from a higher level. In the example of student grades an automatic query engine would need to recognize whether grade A is better or worse than grade B. The common way to compare grades would be by converting to numbers with a mapping on to the reals or integers. The standard computer lexical convention would give the wrong result.

The outcome for this contribution to the debate on theory for data mining is that set theoretic methods are inadequate for reliability when there are potential problems of context involving more than one level. We have already commented on the difficulty of extending the set theoretic relational model with universal relations. This seems a general problem of making set theoretic methods universal. We need to move into constructive mathematics which is represented in the state of the art by category theory. Formal categories and objects, functors and natural transformations handle levels and there are proofs to show that they take the form of universal limits like pullbacks. Category theory has already been applied to object-relational databases [Nelson & Rossiter 1996]. Here, the formalism is extended to cover mining data warehouses. A brief review of category theory, including the construction of pullback diagrams, is given in [Heather & Rossiter 1998].

Exact Data Mining

A pullback diagram can be applied to the universal problem of knowledge discovery. An example of student grades can be given a general formulation by applying Figure 1 shown below. The category names refer to **C** for the candidates, **S** for the subjects and **G** for the grades.

$$
\begin{tikzcd}
 & \mathbf{C} \arrow[dr, "f"] & \\
\mathbf{C}\ X_{\mathbf{G}}\ \mathbf{S} \arrow[ur, "f^{*}(t)"] \arrow[dr, "\in_{f}(t)"'] & & \mathbf{G} \\
 & \mathbf{S} \arrow[ur, "t"'] &
\end{tikzcd}
$$

Figure 1: pullback of t along f.

The arrow f maps each candidate on to one or more grades. t gives the grades which were obtained for a particular subject. This arrow performs the role of insertion of the category of subjects into the category of grades. The pullback (limit) $\mathbf{C}\ X_{\mathbf{G}}\ \mathbf{S}$ is all the candidates in the subjects over grades. It is now easy to interpret the previous statement that f(**C**) corresponds to t(**S**) where f(**C**) and t(**S**) are both objects of **G**. Subject grades correspond to candidate grades: f(**C**) ~ t(**S**).

In general the arrow $f^{*}(t)$ is the functor formally representing the discovery of knowledge in an operational sense. In this example $f^{*}(t)$ is a projection saying how a particular candidate has fared. $\in_{f}(t)$gives how well particular subjects were attempted. Note the difference between these two. The former is a functor providing exact information; the latter is a natural transformation at the message (policy) level. The sense of the commuting diagram is also now clear:

$$f \text{ o } f^{*}(t) = t \text{ o } \in_{f}(t)$$

For consistency in our deductions, the measures of how well a particular candidate has done and how well a particular subject has been attempted are expressed over the same collections of grades.

This example shows well the difference between the use of universal theory in constructive mathematics and the axiomatic set theory style of SQL where a kind of brute-force has to be applied to extract exact knowledge as a member of the powerset. The better scientific approach is to conceptualize from the three-level standpoint of this example.

Application to Data Warehousing

We can extend the power of our approach in a number of ways. Firstly, we can view the arrows $f^{*}(t)$ and $\in_{f}(t)$ as projection arrows π_l and π_r respectively. Secondly, we can show the power of the arrow by representing features of a query language normally represented in calculus form such as the universal quantifier $\forall$. Thirdly, we can type the arrows more strictly if necessary so as to express more constraints in an application. For example they can be typed as *monic* (injective in set) meaning a 1:1 map, *non-monic* meaning a map that may not be 1:1, *epimorphic* (surjective in set) meaning all of the target is assigned, *non-epic* meaning that all the target may not be assigned, *isomorphic* (bijective in set) meaning the arrow is both monic and epimorphic, *partial* meaning the whole of the source may not be assigned or *total* meaning all of the source is assigned. The diagram incorporating these enhanced features is shown in Figure 2.

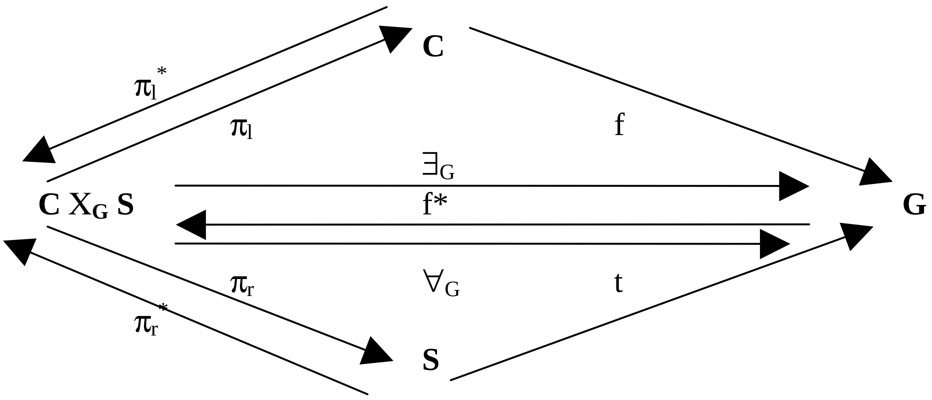

Figure 2: Pullback of t along f showing fuller collection of arrows.

As a starting point, the pullback diagram in Figure 2 gives semantics close to that of an object structural or entity-relationship model. For example the properties of π indicate the participation of **C** and **S** in the subproduct and hence in the relationship. For instance if π_l is not epic, then every member of **C** need not participate in the product **C** X_G **S.** However, the diagram is actually much richer than this also containing the duals to the projection arrows (π_l^*, π_r^*), universal and existential quantifiers ($\forall$, $\exists$) and the pullback functor f^*. The nature of these further arrows, together with those already introduced, is shown in the table in Figure 3.

arrow	type	picks out	for	source	comment
f	monic, partial	G	given	C	grades for a given candidate
t	monic, partial	G	given	S	grades for a given subject
π_l	total	C	given	C X S	candidates taking an exam paper
π_r	total	S	given	C X S	subjects taken in exam papers
f*	epic, partial	C X S	given	G	paper where a particular grade obtained
π_l^*	epic, partial	C X S	given	C	papers taken by a candidate
π_r^*	epic, partial	C X S	given	S	which candidates took each subject
$\exists$	total	G	some	C X S	the grade given to a particular paper
$\forall$	total	G	all	C X S	range of grades obtained on the papers

Figure 3: Table showing purpose of each arrow in Full Pullback Figure

The next task is to create a universal relation across a knowledge base. To achieve this we create a pullback for each relationship in the data warehouse and connect the pullbacks by a technique known as pasting. The resulting universal relation U is a collection of all the attributes, cardinality n, involved in a knowledge base. It is defined intensionally by:

$$U = A_1 \text{ X } A_2 \text{ X}, ..., \text{X } A_n$$

where A_i is an attribute ($1 <= i <= n$).

A number of alternative decompositions can be made of U into various relational schemas R with instances I. These decompositions are projections from the whole product U onto collections of subproducts R so that the j^{th} projection $\pi_{(Rj)}$ can be defined by the arrow:

$$\pi_{(Rj)}: U \rightarrow R$$

These decompositions should be lossless with respect to join dependencies, that is the decompositions can be joined (pasted) together to return U. However, in some knowledge bases, it may not be possible to construct U because design decisions have not produced a lossless decomposition.

In a categorial model, the pullbacks are pasted together as shown in Figure 4 to achieve the universal relation U. The category **T** represents teachers. In the pasted diagram, further diagonal arrows can be added to the diagram which will be natural transformations Φ. Interrogating the knowledge base as a pullback can be performed either by following individual arrows or by composition of arrows within a pullback or across pasted pullbacks. For example, in Figure 4, the composition t o t' o π_r' returns grades obtained in an exam in subjects taught by a particular teacher.

$$\begin{tikzcd}
\mathbf{(C\ X_G\ S)\ X_S\ T} \arrow[r, "\pi_l'"] \arrow[d, "\pi_r'"'] & \mathbf{C\ X_G\ S} \arrow[r, "\pi_l"] \arrow[d, "\pi_r"'] & \mathbf{C} \arrow[d, "f"] \\
\mathbf{T} \arrow[r, "t'"'] & \mathbf{S} \arrow[r, "t"'] & \mathbf{G}
\end{tikzcd}$$

Figure 4: Pasting of Pullbacks of t along f with t' along π_r

An important requirement for mining data warehouses is to consider the powerobjects of data collections rather than just the individual objects. Powerobjects involve taking all possible combinations of the objects in a collection as needs to be done with the student grades to answer certain queries such as: who has passed all their exams? The diagram in Figure 2 can then be re-constructed as in Figure 5 with the nodes as powerobjects as in the logical data model [Kuper & Vardi 1993]. The arrows between the nodes now define rules between clusters of data instances and therefore represent knowledge discovery in a universal manner.

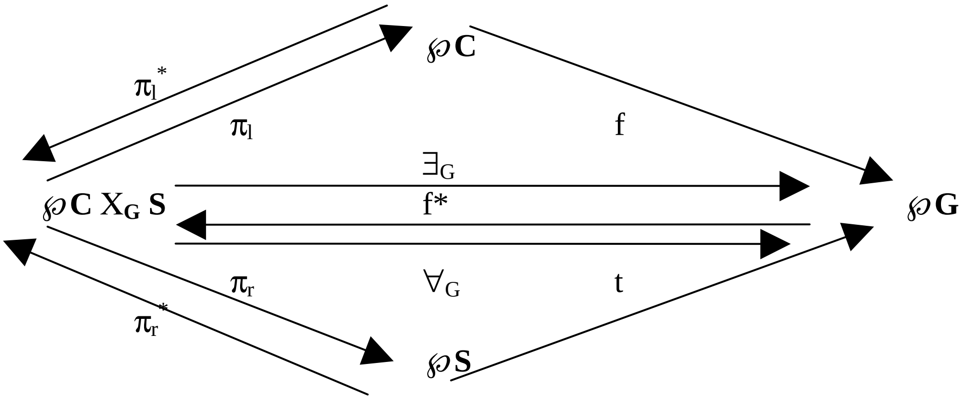

Figure 5: Pullback of t along f with nodes as powerobjects

Finally the pullback diagram is generalized by replacing database classes by whole databases so that it represents the connection of one database to another over a third database. The arrows in these pullbacks are at the data warehousing level, in effect facilitating the pasting of one database to another.

The work demonstrates the potential of category theory for formalising the connections between large-scale heterogeneous systems in a manner that facilitates knowledge discovery.

References:

B.Eaglestone & N.Masood, Schema Interpretation, an Aid to the Schema Analysis in Federated Databases, *Int. CAISE'97 Workshop Engineering Federated Database Systems (EFDBS'97)* 23-32 (1997).

K.M.Kuper & M.Y.Vardi, The Logical Data Model, *ACM TODS* **18** (3) 379-413 (1993).

Colin McLarty, *Elementary Categories, Elementary Toposes*, Oxford Logic Guides 21, Gabbay, D, Macintyre, A, Scott, D, (edd.), Clarendon, Oxford (1995). 265pp. ISBN: 0 19 851473 5.

D.A.Nelson & B.N.Rossiter, Prototyping a Categorical Database in P/FDM, *Proceedings of the Second International Workshop on Advances in Databases and Information Systems (ADBIS'95),* Moscow, Springer-Verlag Workshops in Computing, edd. J.Eder and L.A.Kalinichenko, 432-456 (1996).

M.Roantree & J.Murphy, Using Federated Databases Metadata in the LIOM Project, *Int. CAISE'97 Workshop Engineering Federated Database Systems (EFDBS'97*) 23-32 (1997).