COVER SHEET

CASE-BASED REASONING FOR DATA WAREHOUSE SCHEMA DESIGN

Avin Mathew*, Lin Ma, and V. Lakshmi Narasimhan
Cooperative Research Centre for Integrated Engineering Asset Management
Queensland University of Technology, George Street, Brisbane, Queensland 4001, Australia
a.mathew@qut.edu.au

ABSTRACT

There have been several efforts made in the area of semi-automated techniques and tools for schema design. Most have focused on database schema design, and little work has been done on data warehouse schema design. This paper presents a new approach to data warehouse schema design based on case-based reasoning theory. Case-based reasoning is a problem solving paradigm that involves the development of a solution space to provide a basis for efficient reuse of proven pre-existing solutions. A detailed system design for the application of case-based reasoning to data warehouse schema design is examined in this paper. Two novel contributions of the paper include the matching of cases by data warehouse schema meta-data, and the use of a business context aware ontology to make intelligent suggestions on entities and attributes. These two approaches assist in improving the case matching and adaptation capability in the case-based reasoning system.

KEY WORDS: case-based reasoning, data warehousing, star schema design, context aware ontology

1. Introduction

Data warehousing technologies and methodologies have matured considerably over the past decade. The industry is lucrative, with the average estimated expenditure over the past few years growing to $40.5 billion (Davenport, 2001). As vast budgets are being allocated for data warehousing projects, the ability to streamline processes within a data warehouse implementation can yield significant immediate cost savings for firms. In this paper, we propose a system based on case-based reasoning (CBR) theory to assist in the schema design process. The system is able to automate schema prototyping to consequently reduce the manual design time for data warehouse implementations.

Using CBR for data warehouse schema design is a logical step as much of the data warehouse literature on schema design is based around illustrating concepts through examples. Techniques and methodologies are presented by using real world cases, which the data warehouse community adapts to their situation. Thus using CBR for data warehouse schema design formalises this inherent process.

Along with rapid schema prototyping, the additional benefits from a CBR system can be found in harnessing the experience of previously successful designs for clients in the same or similar industries. Because designs can build upon an existing knowledge base, designers can also propose solutions in domains where the designer is not an expert. While designs will need to be validated for correctness, the approach can provide an initial model which can be further refined.

The rest of the paper is organised as follows: Section 2 provides background information on case-based reasoning and data warehouse schema design; Section 3 presents related research in using CBR for schema design; Section 4 provides a detailed outline of the proposed CBR system; while Section 5 summarises the paper and provides pointers for further research in this area.

* Corresponding author

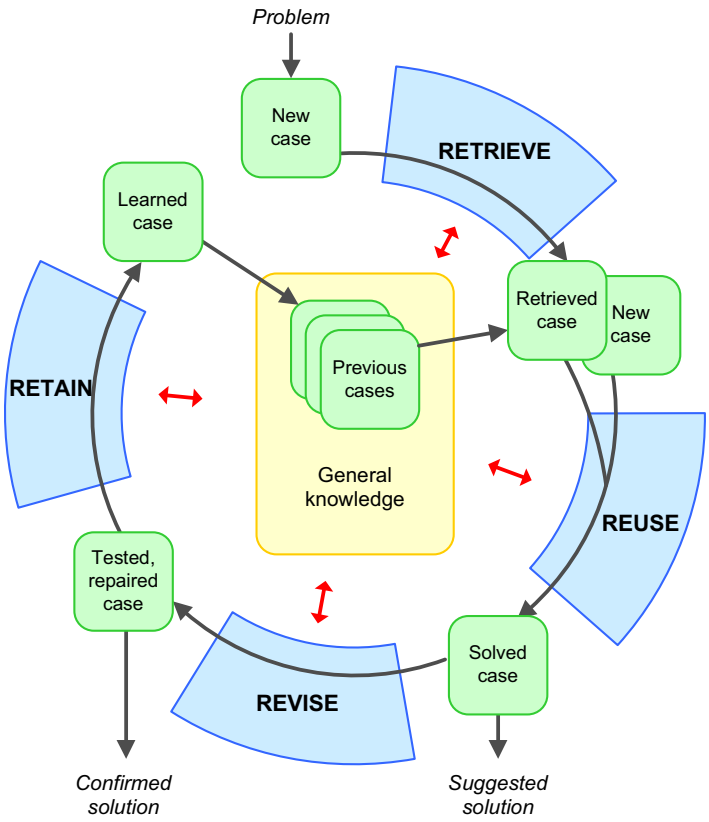

Figure 1: Case-based reasoning lifecycle

2. Background

2.1. Case-based reasoning

Learning through experience is an important approach that humans employ to comprehend new problems. For instance, medical professionals are trained to re-use diagnostic techniques and while prescriptions may require modification to suit particular circumstances, the procedure significantly reduces the amount of work. Sustained learning is a consequence of such reasoning – with successfully solved problems, the experience is retained to solve future problems; with unsuccessful problems, the reason for failure is identified and avoided for future problems.

Case-based reasoning is formed on the above methodology. The CBR lifecycle as described by Aamodt & Plaza (1994) is shown in Figure 1 and involves a reasoning cycle of four processes: (1) retrieving the most similar case/s; (2) reusing the information and knowledge in the case to solve the problem; (3) revising the proposed solution; and (4) retaining the parts of the experience likely to be useful for future problem solving.

The most basic representation of a case is the problem and its corresponding solution. Additional case features can be stored in order to maximise the likelihood of matching the closest case against a given problem during the retrieval stage. A case base or library organises and indexes cases by selected features, and case organisation is designed to achieve two goals: (1) to provide efficient searching during the case retrieval stage, and (2) to properly integrate new cases during the case retaining stage.

2.2. Data warehouse schema design

Decision support systems often form the core IT infrastructure in a business because they give companies a way of turning knowledge into tangible outcomes. The amount of data available in

companies is often overwhelming, and collecting, maintaining, and analysing data requires significant organisational commitment. Many companies have turned to data warehousing to bridge the gap of turning data into knowledge. The data warehouse forms the backbone for informational requirements to decision support systems. A data warehouse serves as an information management solution that integrates information across domains, organisations, applications, and other barriers. It serves as a conduit of accurate and timely information for analysis tools. In effect, it supports decision support systems. Physically, a data warehouse is a data repository devoted to analytical processing, as opposed to an online transaction processing (OLTP) database.

The data warehouse schema is a representation of how data is structured and organised in the data warehouse. The methodologies proposed for data warehouse schema design can be categorised into three groups: user-driven, data-driven, and process-driven (List, Bruckner, Machaczek & Schiefer, 2002). Each methodology differs in the amount of structured information used to build the data warehouse, from less (in the case of user-driven approach) to more (in the case of process-driven approach) respectively. The user-driven approach involves interviewing different user groups to elicit their requirements for the data warehouse; the data-driven approach examines the corporate data model and underlying OLTP systems to determine the applicable transactions; the process-driven approach looks at the business process models to understand the informational requirements for individual processes.

A plethora of data warehouse schema modelling techniques have been proposed by researchers which include, but are not limited to: Application Design for Analytical Processing Technologies (Dulos, 1996), Stars (Peterson, 1996), Dimension Modelling (Golfarelli, Maio & Rizzi, 1998), Object Oriented Multidimensional Modelling (Trujillo & Palomar, 1998), and starER (Tryfona, Busborg & Christiansen, 1999). All of the approaches employ different syntactical representations of the data warehouse schema, while the underlying semantics remain largely the same and hence all approaches are compatible with our proposed system.

3. Related Work

In this section, related applications of case-based reasoning and data warehouse schema design are reviewed. To the best of our knowledge, there have been no attempts in employing case-based reasoning for data warehouse schema design. The closest area is case-based reasoning for database schema design, in which there have been three main efforts.

DES-DS (Design Expert System for Database Schema) – Paek, Seo & Kim (1996) designed the DES-DS with two main components, a Domain Dependent Case Base (DDCB) and a Domain Independent Case Base (DICB). The DICB consisted of nine generalised schema cases that covered different combinations of cardinalities (many-to-many, one-to-many, and one-to-one) and dependencies (partial key, transitive, and full functional). The DDCB contained complete schemas that were hierarchically indexed by a single textual identifier indicating the business area that the schema described. Case representation comprised of the aforementioned business identifier, the schema in the form of a Relation Concept Graph (RCG), and a text description of the schema. Case matching was performed by specifying user requirements in the form of a RCG, and using graph matching techniques. If the CBR system could not match an appropriate case in the DDCB, it would then derive a solution from one of the cases in the DICB.

CSBR (Common Sense Business Reasoning) – Although not based on CBR theory, but more so on methodology, Storey, Chiang, Dey, Goldstein & Sudaresan (1997) introduced a database design system CSBR. Knowledge in the system is divided into three components: an Application Case Base (ACB), an Application Domain Base (ADB), and a Naive Business Model (NBM). Each of these components represents a different layer of abstraction, from the more specific ACB which contains actual cases to more abstract NBM which stores generic business logic. One distinguishing difference compared to the above CBR systems was the use of the NBM as a thesaurus. As user provided terms

for entities and attributes may vary compared to the stored cases, the NBM could resolve user terminology to case terminology.

CABSYDD (Case Based System for Database Design) – Choobineh & Lo (2004) also designed a case-based reasoning system for database schema design named CABSYDD. It also comprised of two components, a CBR system and a module that would derive schema from first principles. The case indexing was similar to that used by Paek, Seo & Kim (1996), in that each schema was hierarchically organised by business area. The hierarchy was formalised by categorising cases using a four tiered structure (sector, subsector, industry group, and department) based on the North American Industry Classification System (NAICS). Case representation included schemas expressed by Extended Entity Relationship (EER) models, textual identifiers for the business area classification, and a textual case description. Matching is performed by calculating the case with the highest matching index score. If no matching cases exist, the system invokes the module that creates a new schema from first principles.

While database and data warehouse schema design are similar areas, there are many differences that distinguish the two. The function and purpose, the data stored, the techniques and technologies used for development, the usage, and the priorities are all different and the differences in attributes have been outlined in many classic data warehouse literature (Inmon, 1995; Kimball, Reeves, Ross & Thornthwaite, 1998).

Outside of CBR, there have been a few systems which try and formalise the data warehouse schema design process. There have been efforts to derive data warehouse schemas from underlying operational database schemas (Böhnlein & vom Ende, 1999; Golfarelli, Maio & Rizzi, 1998; Palopoli, Terracina & Ursino, 2003), from business process models (Böhnlein & vom Ende, 2000; Kaldeich & Sá, 2004), from conceptual graphical models (Hahn, Sapia & Blaschka, 2000), and from XML sources (Golfarelli, Rizzi & Vrdoljak, 2001). None of the systems exploit any knowledge about previous data warehouse implementations, leaving an open area of research into investigating CBR-like systems.

4. Case-Based Approach to Data Warehouse Schema Design

4.1. System Architecture

The CBR system architecture, as shown in Figure 2, uses a split architecture containing a data layer that comprises of a database of cases and business terminology, and a processing layer in which the CBR functionality is implemented.

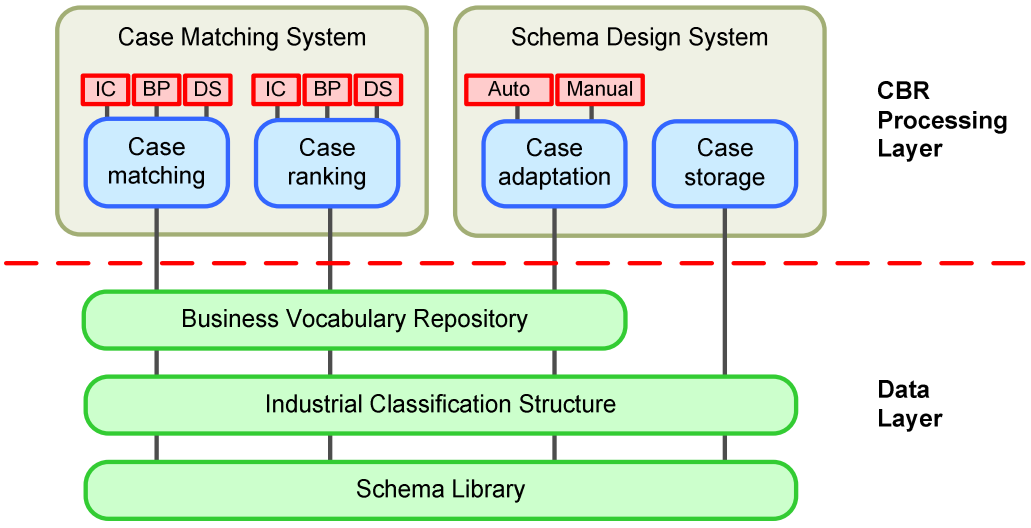

Figure 2: System architecture

- The Schema Library is an organised database of stored cases. Each case is represented by the schema design and associated meta-data.
- The Industrial Classification Structure provides data on organising cases in the Schema Library.
- The Business Vocabulary Repository is a reasoning system that identifies similar business terminology.
- The Case Matching System consists of two elements to cover the first two CBR processes – retrieval and reuse. The system matches user provided terms with case attributes from the case library, and ranks matched cases according to the problem scenario.
- The Schema Design System also consists of two elements that cover the last two CBR processes – revise and retain. The system makes suggestions on the selected schema case, and inserts the modified case in the Schema Library.

Each of the above systems are described in detail in the following sub-sections.

4.2. Schema Library

4.2.1. Case representation

According to Kolodner (1993), a case is "*a contextualized piece of knowledge representing an experience*" and comprises of two characteristics: the solution itself and the context of the solution. The existing research on CBR for schema design (Choobineh & Lo, 2004; Paek, Seo & Kim, 1996; Storey, Chiang, Dey et al., 1997) focussed on the former characteristic, and largely ignored the second.

We show that by capturing the case context, a greater number of indexes can be formed leading to more efficient case matching. The features selected for each data warehouse schema design case attempt to encompass all aspects of the design solution. By developing a comprehensive list of features that represents the totality of the experience, more flexible matching scenarios can be implemented. Feature selection has been approached by structuring the case through a conceptual meta-model for data (Seiner, 2003). The model contains 14 subject areas for meta-data: business function, subject area, purpose, steward, location, community and audience, security, data related, time and date, media type, package, status and version, project and progress, and event. The last two categories are not used for data warehouse schema case representation as they are not relevant.

Under the *Article* category (a category for the primary data in question) is the data warehouse schema with the schema attributes including the entities (facts and dimensions), relationships, relationship cardinalities, attributes, attribute types, and attribute constraints. An important associated category is the *Data related* category, which contains the ETL calculations performed on data sourced from the underlying OLTP system which are necessary to populate each data warehouse fact.

The *Subject area* category contains information on the industrial classification of the business who use the data warehouse. The industrial classification is a hierarchical entity, at the top level indicating the sector of business, and at the bottom level indicating the functional department. This attribute provides support for case organisation and is discussed in Section 4.3. This is opposed to the company name which is found in the *Location* category and is stored for completeness.

Drilling down into the organisation, the related business processes that are described by the schema are identified through the *Business function* category. The business process description may be a single identifier or multi-valued list of processes that the schema supports depending on the level of granularity required. A textual description of the schema outlining its purpose is identified through the *Purpose* category.

Figure 3: Data warehouse schema design case representation

The *Steward, Community and audience*, and *Security* categories list organisational elements, which can consist of individual people, organisational roles, groups of people, or entire organisations. Designer and implementer data are included as they can provide an indication on the quality of the underlying schema. Using these attributes to judge quality through an automatic case matching system is a difficult task and user input will therefore be required. The user roles attribute indicates the organisation role of the users who utilise the data warehouse in their decision making process. Access privileges are stored to be able to mark schemas as classified for security administration.

The *Media type* category covers the support technology in terms of the data warehouse hardware and software environments, and also the OLTP hardware and software environments. This includes the Online Analytical Processing (OLAP) tools used for data exploration. The *Package* category indicates how the case is related to other cases in the library. The project group identifies a collection of cases from a single data warehouse implementation in the organisation. The parent case is used to point to the case from which the current case is derived or reused. The *Status and version* category contain the status of the case, such as “in production”, “test”, “under review”, or “reviewed”, the version, and the language of the case. The *Time and date* category contains lifecycle dates which can be split into two categories: data warehouse lifecycle dates, and CBR lifecycle dates. The design and implementation dates fall under the former category, while the case library entry and last update dates fall under the latter. This category of meta-data provides the case matching system with rankings on timing and popularity of schemas.

The above list of attributes is not exhaustive and they have been selected on the basis of being objective and factual attributes, rather than subjective. Metrics such as quality and risk are arbitrary classifications, making it difficult to maintain consistency between cases. Monetary metrics such as costs or cost savings are influenced by many inter-related assumptions and factors and would skew the results from case matching algorithms. While not all case representation attributes contribute towards automated case matching, they provide information to the user of the CBR system who can manually use such case information.

4.2.2. Case organisation

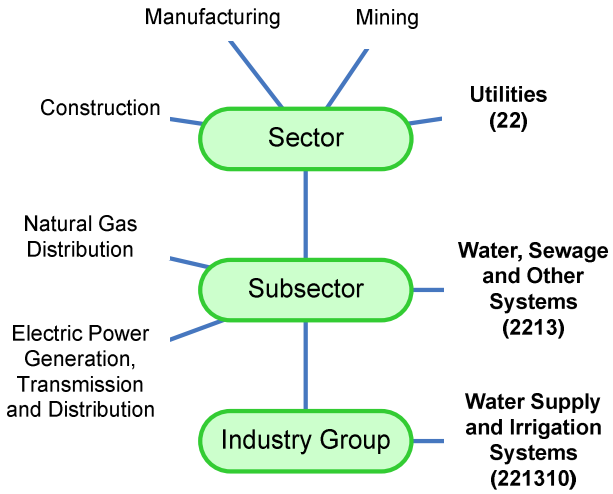

Figure 4: NAICS hierarchical example for Water Supply and Irrigation Systems

Having a set of cases is not sufficient for the retrieval step; the cases must first be organised in a case library. The simplest form of case organisation is having a flat array structure, whereby case matching takes place on a sequential basis. Two other popular techniques for case organisation are hierarchical trees and discrimination networks (Kolodner, 1993). As in the previous cases of case-based reasoning schema design (Choobineh & Lo, 2004; Paek, Seo & Kim, 1996; Storey, Chiang, Dey et al., 1997), the case library is organised through a hierarchical business classification system. The justification underlying the hierarchical organisation structure selection is that from the features identified for case representation, the only suitable ones are the industrial classification and business processes. Many industrial classification taxonomies have been published (Borschiver, Wongtschowski & Antunes, 2004) and having a precompiled structure facilitates the ease of adding new cases to the case library. Figure 4 shows an example of the hierarchal structure from the North American Industrial Classification System (NAICS). Constructing a similar comprehensive taxonomy for business processes would be difficult due to the domain specificity of certain processes, and the steadily increasing number of processes.

The number of hierarchical levels between industrial classification taxonomies typically vary between three and four. However, the number of levels does not preclude any system from being compatible with CBR model. More levels in a system means an increase in the granularity of the case library, which can lead to more accurate case matching, at the expense of servicing additional administrative overheads in case matching and storage due to tree traversal. As industrial classification taxonomies do not include department level information, an additional level can be added to provide a finer grain.

4.3. Business Vocabulary Repository (BVR)

For terminology to be meaningful, it must be used within an appropriate context. For example, a customer in a restaurant environment is equivalent to a patient in a hospital and a guest in a hotel. As discussed by Gust (1991), context independent definitions are infrequent. Hence the Business Vocabulary Repository serves as a controlled vocabulary, or thesaurus, to associate business entities from different contexts. Entities are stored in the BVR using an ontological graph structure. Each node contains the business terminology while edges create semantic associations between each term. Associations are only of one type and connect terms that are equivalent. As the BVR simply serves as a conduit to synonyms, other associations to broader, narrower, related, converse, or homonym terms (Townley & Gee, 1980) are not required.

In order to define a meaningful business context, terminology can be associated with zero or more industrial classification categories. Multiple industrial classifications can be attached to a term, as certain terms can have the same meaning in different contexts. The attached industrial classification/s can then be used during the case matching and automatic adaptation processes to give a distance

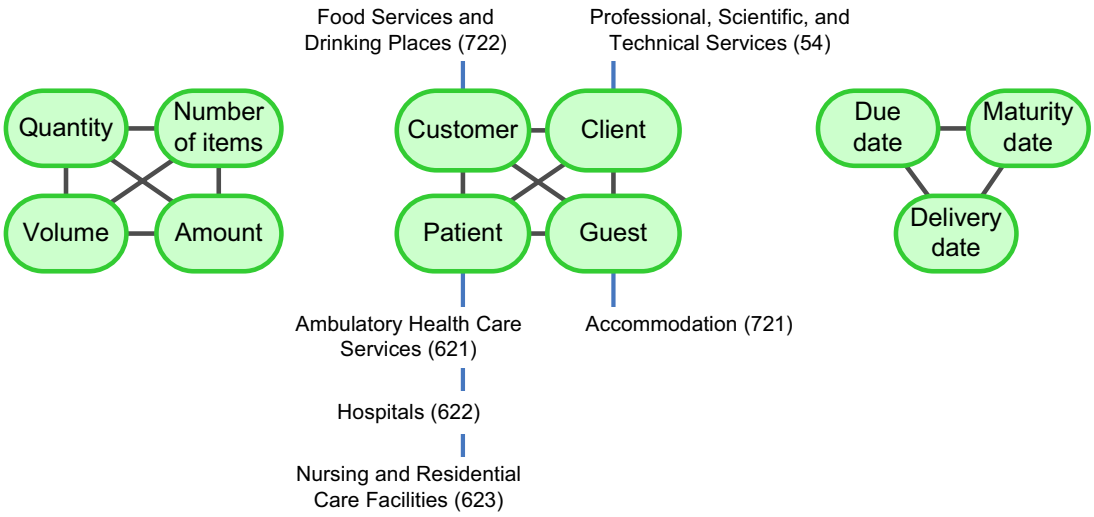

Figure 5: Using industrial classification for business context association

ranking. By calculating the distance between the current and desired context, terms can be ranked in order of (1) matching the industrial classification, (2) matching a classification in the same hierarchy, or (3) matching no classification.

4.4. Case Matching System

The Case Matching System consists of two main components as illustrated in Figure 6: a case matching module and a case ranking module. These two components are described below.

4.4.1. Case matching

In section 2.2, we described the three approaches for data warehouse schema design: user-driven, data-driven, and process-driven. As elaborated below, the case-driven system uses elements of the latter two approaches to enable case matching from three perspectives: industrial classification, business process, and data sources. The three perspectives are not mutually exclusive, but can be used in various combinations to provide a more detailed specification on which cases should be matched. While these three perspectives are not the only parameters that can be matched in the CBR system, they are identified as the most influential ones.

Industrial classification (IC) – Cases in the case library are hierarchically organised by their industrial classification, and it is consequently used as an index. As the IC is hierarchical entity, searching for cases based on their industrial classification can be performed at different hierarchical levels. A specific industry does not need to be targeted and case matching may take place on an entire sector or subsector rather than industry group, which produces more generalised results.

Business process (BP) – The information represented in each case that is relevant to the business process view is the process description and user roles. For example, a designer may search for a particular schema that represents a certain business process, such as a *'pump maintenance'* or *'pipe manufacturing'* process. A designer may also input the roles of users who would use the warehouse. For example, cases could be filtered by the roles of *'electrical technician'* or *'accounts manager'*.

Data source (DS) – Matching schemas by data source gives an indication on potential types of calculations and aggregations that can be performed with the data sets that an organisation already has. This methodology is particularly useful when enterprises have information systems with similar database schemas. For example, an enterprise that uses the same asset management information system as one in the case library can use the same calculations and aggregations, since the data sources

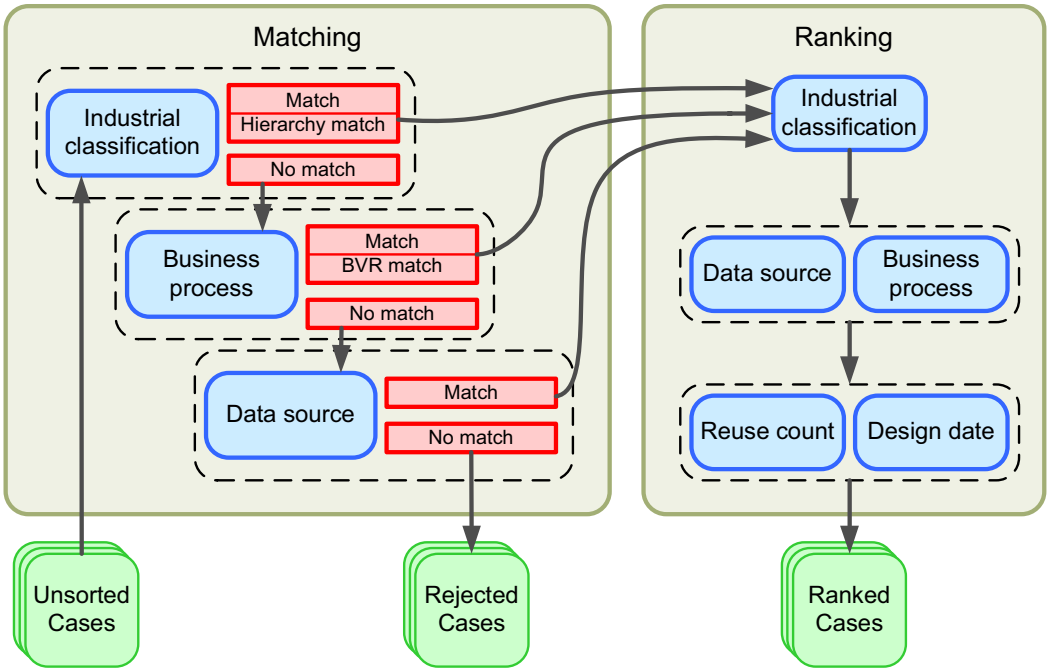

Figure 6: Case Matching System

are the same. Data source information spans ETL calculation, and software environments stored in the case repository.

An issue with both the matching of business process and source system information is matching entities with the same definition. The definition of each industrial classification is rigidly defined by the chosen classification system. However, there are no similar comprehensive enumerations for business processes types or source systems, and descriptions can be subjective. Hence issues arise when similarly named business processes in other organisations are different in meaning. For example, a '*pump maintenance*' process may involve a bearing lubrication step in one company, but the step may be omitted in another organisation. The granularity of data representation plays a significant factor in matching. Increased granularity (e.g. including a description of every step in the business process instead of just the name of the process) will provide more accurate matches, but at the expense of increasing the complexity of the system. The same argument equally applies to data sources – increased granularity would entail the inclusion of schema attributes in the data source in addition to the textual label of the data source.

Matching begins by waterfall filtering unsorted cases according to the industrial classification, business process, and data source. Cases are filtered to eliminate those that have no relevance to the current problem scenario. The specified parameters do not need to match exactly, but can use the hierarchy for IC index or BVR to match for the BP index to improve case generalisation. If the BP index is not specified during case matching, the process becomes an exploratory analysis of potential data warehouse uses in the firm.

4.4.2. Case ranking

The case matching process may identify zero or more cases that are applicable to the problem. In instances where multiple cases are identified, cases can be ranked according to their degree of match. The ranking procedure consists of three different scoring mechanisms. Each step provides a score to determine the case relevance.

The first measure is the industrial classification index. The tree structure can be used to ascertain a ranking score by using the distance from the selected leaf node to the leaf node of a matched case. Hence, those cases in the same branch hierarchy will rank more favourably than those outside the

hierarchy. As ‘relatedness’ between the top level categories in the IC is not represented, all categories outside the hierarchical branch receive the same score.

The second step is using the BP and DS indices to rank the cases. These indices are nominal, using qualitative rather than quantitative values. They cannot be translated to an ordinal scale for ranking, because developing such a scale for each business process and data source is too time-consuming. Instead of an ordinal ranking system, a simplified matching scheme is used to determine whether the BP and DS indices either: match, match a BVR result, or do not match. Those cases that match the BP and DS indices receive the highest ranking score, next are those that required the BVR, and the lowest ranking is given to those that do not match.

Date attributes also influence the ranking score in two ways: preference is given to those cases that have had a higher usage for case derivation and to those cases that are newer. Cases with a higher reuse count indicate a design that is more easily abstracted, and newer designs will typically indicate a greater relevance for modern systems and processes.

4.5. Schema Design System

In a typical CBR approach, the highest ranked case is selected as a base template to derive the design solution for new case. It is unlikely that the selected case exactly matches the requirements of the user and hence the case requires modification.

Schemas can be modified by changing the fact attributes, the dimensions themselves, or the dimension attributes. Extraneous or irrelevant attributes may occur in fact or dimension tables either because the fact metrics were/are not applicable to the targeted users in the decision making process, or because required data sources were/are not available.

The schema design process is divided into two stages: (1) where automatic case adaptation is undertaken to intelligently determine schema modifications from knowledge stored in the Schema Library and BVR, and (2) where manual refinement of the schema is conducted by the schema designer.

4.5.1. Automatic case adaptation

If the matched case is not within the desired industrial classification and business process, then the Schema Library and BVR are consulted to adapt the matched case to a new case. The quantity and quality of data in the Schema Library and BVR will determine the degree of automation possible. Automatic adaptation provides suggestions on modifications which need to be accepted or rejected by the schema designer.

Fact and dimension adaptation
The first stage in automatic adaptation is the examination of similar facts from the same desired business process in closely linked industries. The BVR is consulted to determine similarity between facts through the equivalence associations in the BVR. Whole dimensions can be similarly determined by examining the business processes in close industries and also common dimensions in the desired industry. These two approaches suggest potential dimensions that are frequently used with the desired business process, and the dimensions that are often used in the desired industry.

Terminology adaptation
Additional adaptation takes place by examining the names of the facts and dimensions, and locating the equivalent terminology in the desired business context through the BVR. If an equivalent term exists in the desired business context, then the located term is automatically substituted. If no equivalent terms exist in the desired business context, a list of equivalent terms can be generated ranked by their distance to the current industrial classification and presented to the user in the manual refinement stage.

4.5.2. Manual refinement adaptation

The suggestions made through automatic adaptation are presented to the schema designer for confirmation. To assist the confirmation process, a likelihood score is given to the suggestions based on the commonality for facts and dimensions, and industrial classification distance for terminology.

The knowledge that is used in case adaptation refinement is captured by the case itself and additional mechanisms, such as weighted fact or dimension indices for particular industrial classification, are not required to capture designer knowledge.

4.5.3. Case storage

The process of inserting newly defined cases into the case library is fairly trivial. Case meta-data needs to be ascertained – some can be automatically determined, such as the parent schema for derived cases, and date information; while others must be input manually by the schema designer.

As the Schema Library is indexed by industrial classification, the new case is inserted into the corresponding classification category. The BP and DS indices are then built from the meta-data and stored in the Schema Library.

5. Conclusion and future work

With data warehousing on the forefront of choice for decision support technology, methods must be devised to enhance the data warehousing design process. This paper presents a case-based reasoning approach for data warehouse schema design. Using a tiered architecture, schemas and associated meta-data stored in the Schema Library can be retrieved to aid the schema design process. As part of the case matching and reuse processes, a Business Vocabulary Repository is used to expand the breath of the case searching, and to determine contextualised terminology for schema design.

The contributions of this paper include: (1) the fusion of CBR and data warehouse theory, (2) a unique case representation and indexing methodology, (3) the use of a context aware vocabulary ontology, and (4) automated schema generation using CBR. This research will primarily benefit the data warehousing industry, particularly companies that provide design and implementation services, as they have greater opportunities to construct comprehensive schema libraries. Gains in productivity by data warehouse designers can be realised through a reduction on time spent on design.

Future directions for this research include the implementation of the system to ensure the validity and prove the viability of the proposal. More elaborate techniques using other case indices can be investigated to further automate case adaptation. The use of cases to form a knowledge repository could also be researched. Instead of derivation from an individual parent case, information from a collection of cases can be used to potentially provide a more rounded solution.

References

Aamodt, A. & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications, 7(1), 39-59.

Böhnlein, M. & vom Ende, A. U. (1999). Deriving initial data warehouse structures from the conceptual data models of the underlying operational information systems. In: ACM International Workshop on Data Warehousing and OLAP (pp. 15-21). Kansas City, Missouri, USA.

Böhnlein, M. & vom Ende, A. U. (2000). Developing data warehouse structures from business process models. Bamberger Beiträge zur Wirtschaftsinformatik(57).

Borschiver, S., Wongtschowski, P. & Antunes, A. (2004). A classificação industrial e sua importância na análise setorial. Ciência da Informação, 33(1), 9-21.

Choobineh, J. & Lo, A. W. (2004). CABSYDD: Case-based system for database design. Journal of Management Information Systems, 21(3), 281-314.

Davenport, T. (2001). Data to knowledge to results: The data context - turning raw data into gold. Boston, Massachusetts, USA: Accenture.

Dulos, D. (1996). A new dimension. Database Magazine, 19(3), 32-37.

Golfarelli, M., Maio, D. & Rizzi, S. (1998). Conceptual design of data warehouses from E/R schemes. In: Hawaii International Conference On System Sciences (pp. 334-343). Kohala Coast, Hawaii, USA.

Golfarelli, M., Rizzi, S. & Vrdoljak, B. (2001). Data warehouse design from XML sources. In: ACM International Workshop on Data Warehousing and OLAP. Atlanta, Georgia, USA.

Gust, H. (1991). Representing word meanings, Springer-Verlag GmbH: 127-142.

Hahn, K., Sapia, C. & Blaschka, M. (2000). Automatically generating OLAP schemata from conceptual graphical models. In: ACM International Workshop on Data Warehousing and OLAP. McLean, Virginia, USA.

Inmon, W. (1995). What is a data warehouse? Prism Tech Topic, 1(1).

Kaldeich, C. & Sá, J. O. e. (2004). Data warehouse methodology: A process driven approach. In: Advanced Information Systems Engineering (pp. 536-549). Riga, Latvia: Springer-Verlag GmbH.

Kimball, R., Reeves, L., Ross, M. & Thornthwaite, W. (1998). The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses. New York City, New York, USA: John Wiley and Sons.

Kolodner, J. L. (1993). Case-based reasoning. San Mateo, CA: Morgan Kaufmann Publishers.

List, B., Bruckner, R. M., Machaczek, K. & Schiefer, J. (2002). A comparison of data warehouse development methodologies - Case study of the process warehouse. In: Database and Expert Systems Applications (pp. 203-216). Aix-en-Provence, France: Springer-Verlag GmbH.

Paek, Y. K., Seo, J. & Kim, G. C. (1996). An expert system with case-based reasoning for database schema design. Decision Support Systems, 18(1), 83-95.

Palopoli, L., Terracina, G. & Ursino, D. (2003). DIKE: A system supporting the semi-automatic construction of cooperative information systems from heterogeneous databases. Software: Practice and Experience, 33(9), 847-884.

Peterson, S. (1996). Stars: A pattern language for query optimized schema. Accessed: 7/1/2005. http://c2.com/ppr/stars.html

Seiner, R. S. (2003). A conceptual meta-model for unstructured data. Pittsburgh, Pennsylvania, USA.

Storey, V. C., Chiang, R. H. L., Dey, D., Goldstein, R. C. & Sudaresan, S. (1997). Database design with common sense business reasoning and learning. ACM Transactions on Database Systems, 22(4), 471–512.

Townley, H. M. & Gee, R. D. (1980). Thesaurus-making: Grow your own word-stock. London: Andre Deutsch.

Trujillo, J. & Palomar, M. (1998). An object oriented approach to multidimensional database conceptual modeling. In: ACM International Workshop on Data Warehousing and OLAP. Washington, D.C., USA.

Tryfona, N., Busborg, F. & Christiansen, J. G. B. (1999). starER: A conceptual model for data warehouse design. In: ACM International Workshop on Data Warehousing and OLAP. Kansas City, Missouri, USA.