Discovery of Generic concepts from Heterogeneous Clinical Information Systems

Uma Srinivasan Anne H.H Ngu Tom Gedeon

School of Computer Science & Engineering
University of New South Wales
P.O.Box 1, Kensington 2033
NSW, Australia.
e-mail: {uma, anne, tamas}@cse.unsw.edu.au
Phone: (612) 385 3970 Fax: (612) 313 7916

Abstract

Most heterogeneous Clinical Information Systems share a strong semantic resemblance in spite of their autonomy and differences in data requirements and design. This semantic resemblance can be exploited when performing schema integration. We propose a methodology to identify a set of generic concepts based on their semantic similarity using qualitative parameters such as entities' usage patterns, database structures and users' domain knowledge. This is different from the traditional data mining methods which have to use the data values of the entities. The generic concepts can be seen as a customized schema which is geared to address different interpretations of the data by different groups of users in a clinical environment.

1. Introduction

A large organization such as a hospital usually has a number of independent Information Systems developed by different departments over a period of time. Typically these are systems that cannot be disturbed and form what are referred to as legacy information systems [Bro92]. In order to obtain any form of integrated information from these heterogeneous autonomous systems, currently the users view the data from each system separately, or use the physical medical record document. This is time consuming and often does not yield the required information. To obtain a holistic picture of a patient's clinical condition, users need to be provided with a seamless integrated view of all the existing Clinical Information Systems (CIS), similar to the HealthCare Cooperating Information Systems (HCCIS) mentioned in [Bro92]. Cooperative information systems typically involve integrating distributed information sources which span both database and Knowledge-based system domain and which may employ heterogeneous data/knowledge representations[PLS92].

CIS users such as doctors, nurses and other health professionals are equipped with a lot of specialised domain knowledge. Although each individual's expertise provides them with a different world view of data, they do share a minimum vocabulary common to the health profession. This knowledge gets represented in the local CISs at the time of specifications, in the form of rules and constraints on data domains and data structures. As every CIS stores patient related data, there is a lot of semantically similar

types of information. This was confirmed by a manual empirical study, which consisted of schema comparison and analysis of three existing CISs in a leading hospital [SNG93].

The schema comparison and analysis leads us to the following conclusions:

- There are a variety of schema conflicts that cannot be resolved syntactically, and therefore the required integrated schema cannot just be a syntactically integrated schema.

- A large amount of knowledge such as rules and constraints on data-domain, data-structures, and so on are inherent in the local schemas. It may be possible to extract this knowledge from the existing databases and used to provide integration.

- There exists a lot of semantic similarity amongst the objects/entities across CISs. The objects/entities in the local schemata can be grouped into a common set of Generic concepts.

- Data held in a particular CIS can play multiple roles depending on the user, and the user can play different roles while using a particular piece of data.

The results of the study also indicate that nearly every Generic Concept that could be manually identified has representative entities from the local CISs. Our approach to integration, therefore, is based on identifying some common generic concepts that exist across these systems, are well understood, and coincide with the users' minimum vocbulary.

Traditional schema integration work focussed on determining similarities based on common attributes [SL90][HR90]. However, it was pointed out that such attribute relationships do not characterise real world systems[SG89].

We propose a methodology that utilises users' contextual knowledge, usage patterns and existing data semantics from local systems to derive these Generic Concepts. We then briefly outline the applicability of the Generic Concepts in providing a flexible user oriented query mechanism. Details of the query mecahnism have been extensively dealt with in [LNS93]. The focus in this paper is on discovering the Generic Concepts(GC).

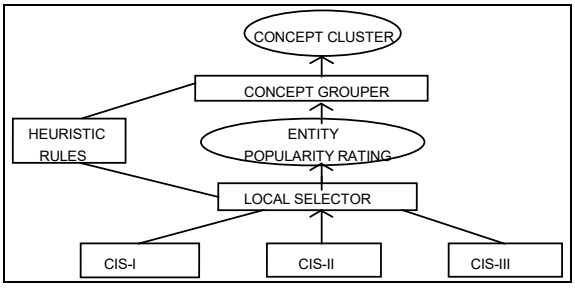
2. Generic Concept Discovery Process

As each department specifies their data requirements independently, the design of each CIS is different. CISs implemented on different environments at different points of time also contribute to heterogeneity and autonomy. Once the systems are implemented, they are managed locally, which results in database evolution over a period of time. This autonomy poses a major problem for any kind of static integration strategy and precludes the use of a conventional query language like MSQL [Lit et al. 87], which can only be used for a pre-defined global integrated schema. Knowledge based integration methods such as [FN92] present an approach to integrate schemas utilising fuzzy real world knowledge. However our approach is geared towards discovering concepts well known to the users of the application domain.

The main motivation in discovering the Generic Concepts (which coincide with a typical hospital user's minimum vocabulary), is to use them as the foundation for building a flexible user-oriented query mechanism. We believe that a user-oriented query mechanism is needed to deal with the complexity of real world cooperating clinical databases. This query mechanism is geared to address different contextual interpretation of the data by different groups of users.

A variety of knowledge discovery methods have been proposed to extract knowledge from existing databases [SF91]. These methods can be broadly classified into two groups: quantitative and qualitative. The quantitative methods use statistical techniques to anlyse data values and create some statistical characterisations [Quin90][ZB90]. The qualitative methods are domain knowledge driven and discover patterns and concepts from existing databases [Fraw90]. However, these qualitative methods still depend on data values to discover patterns and concepts.

While the approach proposed here is qualitative in nature, the resources used for concept discovery are quite different from the usual data mining approach. We use four important parameters that are available in existing CISs to discover knowledge. These are: (i) usage patterns of entities/objects of a set of semantically similar CISs, (ii) current access patterns of different types of users, (iii) existing database structures, and (iv) existing domain definitions. It is not necessary in this approach to look at individual data values.



Entities of each CIS are assigned a *Usage-frequency* rating based on the information obtained from their respective transaction log files (which are set up in each CIS to log an entry each time an entity is used; either directly or indirectly while participating in a relationship with other entities). **Users** of each CIS are assigned an *Access-frequency* rating based on their use of existing CISs, (which is obtained from the user log files).

The two main components of the Concept Discovery algorithm are: the Local Selector and Concept Grouper. The Local Selector assigns a *Popularity-rating* for each entity in a CIS based on the *Usage-frequency* of **entities** and the *Access-frequency* of **users**. The Concept Grouper uses a set of heuristic rules to group the (popularity) rated entities into concept clusters. Popularity is a useful measure to group concepts with, as it indicates that if a concept is popular, it is perhaps used by more peple, more often.

The interactions among these components are as shown in Figure 1. The rectangles indicate the functional modules and the ovals indicate the output of these modules.

Figure 1: Components of Concept Discovery Algorithm

The functions of the Local Selector and Concept Grouper are described in detail in the following sections.

2.1 Local Selector

Each CIS has a set of hospital users who use the information based on their own specific requirement. The users' information needs varies for each group of users. This is mainly due to the role each user plays in the hospital. Hospital users can be classified into fairly distinct groups; such as Registrars, Specialists, Nurses and so on. These users are governed by their contextual knowledge while looking for information. The Local selector classifies entities of each CIS based on their usage by different groups of hospital users, in such a way that some conceptual cohesiveness can be established across different CISs. Table I is an indication of how the users of different CIS use the entities of the respective CIS.

Column 1 represents the different user-types in the application environment. Columns 2, 3 and 4 indicate a representative list of entities used by different user-types in three different existing systems. (This is a representative sample from three real life systems and the entries under each CIS are actual names of entities as they appear in their respective CISs).

Table I: Usage pattern of CIS users

User-type	**CIS-I (Radiotherapy)**	**CIS-II (Intensive-care)**	**CIS-III (Diabetes)**
Doctor (GP/Registrar)	PMI Cln-msrmnt Treatment Tumour	Patient Event Therapy	Patient Cln_assesment Drug_dosage Drug-trtmnt
Specialist (local to CIS)	PMI Treatment Tumour	Patient Therapy Admission_source	Patient Cln_assesment Drug_dosage
Specialist (external to CIS)	PMI Tumour PMI_history	Patient Thearapy Event	Patient Drug_trtmnt
Nurse	PMI Cln_msrmnt Treatment	Patient Event Therapy	Patient Drug_dosage
Dietitian	PMI Treatment	Patient Therapy	Patient Drug-trtmnt
Physiotherapist	PMI Treatment	Patient Thearpy	Patient Cln_assesment Drug_trtment

The Local Selector identifies the popular entities amongst the hospital users within each CIS and assigns a *popularity-rating* for each entity. This is a three-stage process: first the *usage-frequency* of the different entities are classified within each CIS, secondly the *access-frequency* of the different types of users are classified, and finally these two pieces of information are combined to arrive at an entity's *popularity-rating.*

The various terms used are defined here:

> *User-type* refers to a member of the set of hospital users called *USERS.*
> *USERS* ==={Registrar, Specialist, Nurse, Dietician, Physiotherapist,...}
> *User-type* ∈ *USERS*

Usage-Frequency refers to the frequency of use of each entity in each CIS. It is a value assigned to each entity of each CIS regardless of who the user is. *Usage-frequency* can take on one of the following values: {**High, Medium, Low**}. The entities of a CIS are divided into 3 groups based on actual usage values obtained from the log files. The top one third are assigned a **High** value implying they are used most often, the next one third are assigned a **Medium** value, and the remaining are assigned a **Low** value. The fuzzy scale of values is used instead of numeric values to stress the qualitative rather than the quantitative nature of our method.

> A total function *usage* assigns a *Usage-frequency* value to each *Entity*
> *usage: Entity* → *Usage-frequency*

Access-frequency refers to the frequency of use of a particular CIS by a particular *user-type.* It is a value assigned to each *user-type* for each CIS. The same *user-type* may have different values of access-frequency for different CIS. *Access-freqeuncy* can take on one of the following values: {**High, Medium, Low**}. (It is to be noted that these values are not in any way related to the previous values.). A high value of *Access-frequency* for a user-type indicates for example that the particular user-type uses that CIS more than once a day, a medium value indicates that, that user-type uses the CIS not more than once a day, and a low value indicates that the user-type uses the CIS less than once a day.

> A function *access* assigns an *Access-frequency* value to each *User-type* for each CIS.
> *access: User-type* → *Access-frequency*

Popularity-rating as the name suggests is an indicator of how popular an entity is within a given CIS. The highest *Popularity-rating* is assigned to the entities being used most frequently by maximum users in each CIS. *Popularity-rating* is a function of *Usage-frequency* and *Access-frequency*.

Popularity Rating is represented on a 3 point scale. The rules to assign *Popularity-rating* are as follows:

Popularity-rating is **High** when	*Usage-frequency* is **High** and *Access-frequency* is **High**.
	Usage-frequency is **High** and *Access-frequency* is **Medium.**
	*U*sage-frequency is **Medium** and *Access-frequency* is **High**.
Popularity-rating is **Medium** when	*U*sage-frequency is **Medium** and *Access-frequency* is **Medium.**
	Usage-frequency is **Low** and *Access-frequency i*s **High.**
	Usage-frequency is **High** and *Access-frequency* is **Low**
Popularity-rating is **Low** when	*Usage-frequency* is **Low** and *Access-frequency* is **Low**.
	Usage-frequency is **Medium** and *Access-frequency* is **Low.**
	Usage-frequency is **Low** and *Access-frequency* is **Medium**.

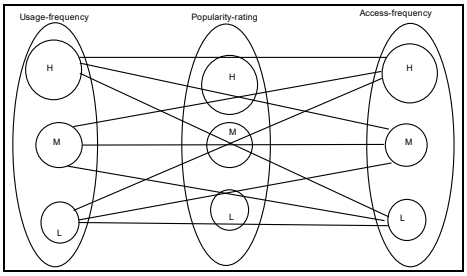
A function *assign* assigns a *popularity-rating* to each *entity* of a given CIS based on the above rules.

assign: Entity → Popularity-rating

Figure 2 shows the relationship between *Usage-frequency, Access-frequency* and *Popularity rating.*

Figure 2: Usage pattern of Entities

The *Popularity-rating* scale is a fuzzy scale that can be controlled by specifying the size of the scales of *Usage-frequency* and *Access-frequency.* *Popularity-rating* is used as the basis for grouping by the Concept Grouper.



2.2 Concept Grouper

The Concept Grouper applies a set of heuristic rules to the entities classified by the Local Selector to create common concept clusters across CISs. The heuristic rules for Cluster classification are based on different possible combinations of values of the following parameters:

- Popularity-rating of entities as assigned by the Local Selector,
- Participating Relationships of entities that are being compared, and
- Domain of the key attributes of entities that are being compared.

Popularity-rating (PR) is the rating given to each entity by the Local Selector. It is expressed on a three point scale, represented by **High (H), Medium (M) and Low(L)**.

Relationship (RE) refers to the group of entities that normally associate with each other (for example, in the form of JOIN operations), during a query formulation process in the existing CIS. This parameter is chosen on the basis of our manual observation which shows that groups of similar entities across CISs participate in similar relationships.

Relationship (RE) is expressed on a three point scale. The heuristics for determining the scale are:

RE is **High** when the the two entities being compared participate in same relationships.

RE is **Medium** when there is partial match in the participating relationships of the two entities being compared.

RE is **Low** when there is no match in the participating relationships of the two entities being compared.

Domain Match (DM) of key attributes indicate that there could be some semantic similarity among two entities being compared. Domain match (DM) of key attributes is expressed on a two point scale.

DM is **M** (match) indicates that the domains of the key attributes of the two entities being compared match.

DM is **X** (no match) indicates that the domains of the key attributes of the entities being compared do not match.

The main heuristic used is that when PR, RE and DM have the same values for the entities of different CIS being compared, then the entities belong to the same concept cluster. If the three values are totally different, then they belong to different concept clusters. Other variations represent different levels of matches when being clustered together.

Figure 3 indicates the type of matches that are performed between the entities being compared. Only representative examples are shown, in order to avoid cluttering the diagram with lines.

The left block represents the PR values that the two entities being compared can take on. **High-Medium** indicates that of the two entities being compared, one has a PR value **High** and the other has a PR value **Medium**.

The right block indicates the results of the comparison of the two entities. **Same** indicates that the 2 entities belong to the same cluster, **Differen**t indicates that they belong to different concept clusters. The other entries indicate intermediate values, for example, **Similar** indicates that they belong to similar concept clusters, but are not the same.

The lines show the relationship among the three parameters when they are compared and show the possible outcomes (Result) of the comparison. The first character annotating the line indicates the value of RE , and the second character indicates the DM value. For example, HM indicates that the Relationship comparison has a High (H) value and the Domains match (M).

The results of the matches can take on a range of values as shown in the right hand block of the diagram.

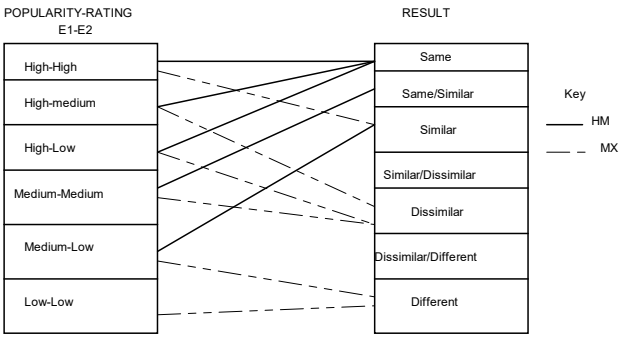


Figure 3: Concept clustering Process

The mappings shown in the diagram can be represented by the following heuristic rules:

If (RE = H) ∧ (DM = M) ∧ ((PR1 = H ∨ PR1 = M) ∧ (PR2 =M)) **then** Result = Same

If (RE = H) ∧ (DM = X) ∧ (PR1 >= M ∧ PR2 = H) **then** Result = Same/Similar

If (RE = H) ∧ (DM = M) ∧ (PR1 = H ∧ PR2 = L) **then** Result = Similar
If (RE = M) ∧ (DM = M) ∧ (PR1 = H ∧ PR2 >= M) **then** Result = Similar
If (RE = M) ∧ (DM = X) ∧ (PR1 = H ∧ PR2 = H) **then** Result = Similar
If (RE = H) ∧ (DM = X) ∧ (PR1 = H ∧ PR2 >= M) **then** Result = Similar

If (RE = H) ∧ (DM = X) ∧ (PR1 <= M ∧ PR2 = L) **then** Result=similar/Dissimilar

If (RE = M) ∧ (DM = M) ∧ (PR1 <= M ∧ PR2 <= M) **then** Result = Dissimilar
If (RE = M) ∧ (DM = M) ∧ (PR1 = H ∧ PR2= L) **then** Result = Dissimilar
If (RE = M) ∧ (DM = X) ∧ (PR1 >= M ∧ PR2 < =M **then** Result = Dissimilar

If (RE = M) ∧ (DM = X) ∧ (PR1 <= M ∧ PR2= L) **then** Result = Different

In the first instance, these rules are applied across two CISs to cluster the entities. Subsequent matching and clustering of other entities from different CIS, are done as follows:

E1 + E2 = Same
implies that E1 and E2 belong to the same concept cluster C1.

When a new entitiy E3 is to be considered for clustering, it is compared to E1 and E2. It belongs to the same cluster as E1 and E2, if it has a result = 'Same' when compared to any one of them, and has atleast a 'Similar' result with the other.

The general rule is
E3 ∈ C1
if ∃ Ei : Ei + E3 =Same ∧ ∀ Ej ≠ Ei, Ej + E3 >= Similar

Thus all the entities are grouped into concept clusters which will be used as the basis for providing a flexible user-oriented query mechanism outlined in the next section.

3. Application of Generic Concepts in Query handling

The important aspect of this query mechanism is that it can be customised to the needs of different types of users, based on their domain and contextual knowledge [LNS93] [SNL94]. The contextual knowledge here refers to the minimum voacbualry of a particular user-type. Each type of user is represented by a User Object.

The User Object consists of two main parts:

- A set of Generic Objects (discovered using the above process), which provides the customised schema for querying in a fashion specific to the individual user's expectations. Each User object has a different composition of Generic Objects, thereby supporting the expectation differences among different users. For each Generic Object, a set of extractors are attached that specify how the data is retrieved and what form of display is required. Each Generic Object is customized to the need of a user by selecting a suitable subset of extractors.
- The Context labeller supports the interpretation differences amongst different types of users. It stores and provides appropriate labels for the information need of a particular user. For example, nurses might interpret a Generic object as "Diet" while the Dietician might interpret the same object as "Treatment". The labeller puts the data in the right perspective for the user. This is similar to mapping the data retrieved from different CISs to the correct context of the user.

An important advantage of such a user-oriented querying mechanism is that we can avoid the process of semantic resolution. The user is the one who decides the semantics of the required data. In a hospital environment, especially in regard to disgnosis and treatment information, only the user (such as a medical specialist) is able to resolve the semantic conflicts through his/her domain knowledge.

4. Conclusion

Schema Integration is an important issue that needs to be addressed when dealing with the heterogeneous Clinical Information Systems (CIS). Most of the work in the literature has focused on structural and/or semantic integration [SLCN88] [SCG92].

An emprical study of three different CIS systems in a hospital revealed the fact that in spite of the autonomy and dynamic evolution of each CIS system, there is an underlying vocabulary that is prevalent across the different systems. A number of words such as Diagnosis, Drug, Patient, Treatment etc are present in some (synonymous) form in every clinical database system. It is the presence of these keywords that led us to look for semantic similarity among the databases. Thus our approach to integration is based on identifying some common generic concepts that exist across different CIS systems, are well understood, and coincide with the users' minimum vocabulary.

We propose a methodology (concept discovery algorithm) that uses the user's context knowledge (the set of user vocabulary), usage patterns of entities and database structures of the local systems to discover the generic concepts. This is a qualitative data mining method which does not need to use any data values of the entities.

The main components of our concept discovery algorithm are the local selector and the concept grouper. The local selector assigns a popularity rating for each entity of a CIS based on usage-frequency and the access-frequency. The concept grouper uses a set of heuristic rules we have developed here to cluster entities that are semantically similiar.

We also briefly illustrate the importance of generic concept in the building of a flexible user-oriented query mechanism in heterogenous CISs environment.

References

[Bro92] The promise od distributed computing and the challenges of legacy information systems. In IFIP DS-5 Semantics of Interoperable Databases Systems: Lorne, Australia, November, 1992.

[FN92] P.Fankhauser and E.J.Neuhold. Knowledge based integration of heterogeneous databases. In IFIP DS-5 Semantics of Interoperable Databases Systems: 150-170, 1992.

[Fraw 90] Frawley W.J. Using functions to Encode Domain and Contextual Knowledge in Statistical Induction, Knowledge Discovery in Databases, AAAI Press, 1990.

[HR90] S.Hayes and S.Ram. Multi User View Integration System (MUVIS): An expert System for View Integration in Proceedings of the 6th International Conference on Data Engineering, February 1990.

[LNS93]N.Lee, A.A Ngu and U.Srinivasan. A Hypertext approach to Querying Clinical Multidatabases. In Proceedings of the 5th International Hong Kong Computer Society Workshop on Next Generation Databases Systems, Honk Kong, February, 1994.

[Lit et al.87] MSQL: A multidatabase language. Tech Rep. 695, INRIA, BP 105,78153 Le-Chesnay, France.

[PLS 92] Mike . P. Papazoglou, Steven C. Laufmann, Timos K. Sellis. An organizational framework for Cooperating Intelligent Information systems. International Journal of Intelligent and Cooperative Information Systems, Vol.1,No.1 91992) 169-202.

[Quin 90] Quinlan J.R. 1990. Learning Logical definitions from Relations. Machine learning, 1(1): 81-106.

[SCG92] F.saltor, M.G. Castellanos and M. Garcia-Solaco. Overcoming Schematic Discrepancies in Interoperable Databases. In IFIP DS-5 Semantics of Interoperable Database Systems, 272-301, 1992.

[SG89] A.Sheth and S.Gala. Attribute Relationships: An impediment in automating Schema integration. NSF Workshop on heterogeneous databases Systems, Chicago, December 1989.

[SF90] L. Shapiro and J. Frawley. Knowledge Discovery in Databases. AAAI Press/MIT Press.

[SK92] A.Sheth and V.Kashyap. So far (Schematically), yet so Near (semantically). In IFIP DS-5 Semantics of Interoperable Database Systems, 272-301, 1992.

[SL90] A.Sheth and J.Larson. Federated database systems for managing distributed, heterogeneous and autonomous databases. In ACM Computing Surveys, Vol 22(3), September 1990.

[SLN88] A. Sheth, J.Larson, A.Carnellio and S. Navathe. A Tool for integrating conceptual schemas and user views, Proceedings of the Fourth International Conference on Dta Engineering, Los Angeles, CA, February, 1988.

[SNG93] U. Srinivasan, A.H.H. Ngu and T. Gedeon. Multilevel Lateral Schema Translation in Heterogeneous Clinical Databases. Tech Report, School of Computer Science and Engineering, University of New South Wales, Australia.

[SNL94] U.Srinivasan, A.A Ngu and N.Lee. A User Oriented Query Mechanism for Clinical Multidatabases. Working Papaer, School of Computer Science and Engineering, University of New South Wales, Australia.

[ZB90] Zytkow J M. and Baker J. Interactive mining of Regularities in Databases. Knowledge discovery in Databases, AAAI Press, 1990.