A Framework and Implementation of *Information Content Reasoning* in a Database

[1]XIAOCHUAN WU and [2]JUNKANG FENG*
[1,2]Database Research Group
School of Computing
University of the West of Scotland
High Street, Paisley PA1 2BE, UK
{xiaochuan.wu; junkang.feng}@uws.ac.uk
[2]Business College
Beijing Union University
No. A3 Yan Jing Dong Li, Beijing 100025, CHINA

*The contributions of both authors are equal. Junkang Feng is the corresponding author.

Abstract: - Databases' capability is limited in terms of inference. Especially, when users explore information beyond the scope of data within databases, the databases normally cannot provide the information. The underlying reason of the problem is that queries are answered based on a direct match between a query and data (up to aggregations of the data). We observe that it is possible to find information from a database beyond that. To this end, we propose a framework for *information content reasoning* in a database. A number of basic concepts are defined first. Then we present the framework and explain how it works. Moreover, we describe how such a framework is implemented by means of a prototype including a test with sample queries.

Keywords: - Information content, Reasoning, Knowledge discovery from databases, Semantic theory of information, Databases

1. Introduction

Database systems store data [7]. Users query a database [2] and a query can only be answered through a 'direct match' between the selection criteria within a query and data (up to aggregations of the data). In a case of querying a database beyond this, the system is unlikely to answer the query. A conventional query is, in essence, concerned with only the *propositional content* of data. Data carry information [6], [3]. A piece of data may carry information about another [1], [15]. It would seem desirable and possible to capitalize on this phenomenon so that more information can be derived than using conventional queries.

Information systems are constructed for storing and providing information. And yet, it would appear that the notion of 'information content' of an information system is elusive. In the field of databases, the information content of a database has been taken as the instance of a database and the information capacity of a data schema as the collection of instances of the schema [9], [10], [11]. Another view on the topic of the relationship between information and data is that if it is truthful, meaningful data is semantic information [8]. We argue that such views miss two fundamental points. One is a convincing conception of 'information content'. To equate data with information overlooks the fact that data in a database is merely raw material for bearing and conveying information.

Information must be veridical (p.10 of [1]), that is, it must relate to a contingent truth [8], while for data there is no such requirement. The other is a framework for reasoning about information content to reveal hidden information. In addressing this problem, our purpose is to look at the relationships between information content, database structure and business rules, and thus discover how tacit business knowledge can then be explicitly expressed and used.

In this paper, we present a novel framework for reasoning about the information content of data in a database. It helps a database system improve its capability of inference. This is achieved by introducing a variety of information sources such as domain knowledge. With the help of outside information sources, not only more queries that deal with a wider range of information than the propositional content of data within database can be answered, but also hidden information within the database system can be discovered. The underlying thought of the framework is based on a concept of *information content*. Fred Dretske [3] firstly introduced the concept. Then Xu, Feng and Crowe [16] extend Dretske's idea and give a more detailed definition of Information Content of states of affairs. Our thoughts are based on the latter definition.

The next section gives some fundamental concepts. Then the framework and a prototype of implementation are presented in the third section. The last section concludes the paper.

2. Basic Concepts

A number of fundamental notions are defined in this section and they are the cornerstones of this paper.

2.1 Information Content

Fred Dretske [3] gives the definition of information content as follow:

"A state of affairs contains information about X to just that extent to which a suitably placed observer could learn something about X by consulting it."

Then he formalizes the above as

"Information Content: A signal r carries the information that s is F = The conditional probability of s's being F, given r (and k), is 1(but, given k alone, less than 1)."

Note that k stands for prior knowledge about information source s.

Here is an example: That John is awarded a grade 'A' for his Programming course contains the information that he has gained 70% or above for that course. The definition above will be used as a cornerstone to define the theoretical foundation of the framework.

2.2 Random Events

Following [16], the definition above is based upon the notion of probability ([1], pp.14-18), and it is strongly connected with the notion of *random event*. Thus, they firstly defined *random event* as follows:

"Let *s* be a selection process under a set C of conditions, O the set of possible outcomes of *s*, which are called states, and E the power set of O, X is a random event if E∋X and there is a probability of X, i.e., P(X). "

For example, to select a student record from Students table randomly in database and it is of a particular student is a random event.

In addition to the definition above, they unveiled a definition of probability space to explain what mean by 'probability distribution' as below:

"Let s be a selection process under a set C of conditions, O the set of possible outcomes of s, E the power set of O and E∋Xi for i = 1,…, n, P_s is the probability space of the random events X_i for i = 1,…, n if $P_s = \{P(X_1), P(X_2),…, P(X_n)\}$ and $\sum P(X_i) = 1$. "

2.3 Random Variables

A random variable is a variable that can hold one of a number of possible values at a time and which one of the values to be hold is determined randomly. For example, as in the above example, Students table contains attributes such as ID, Name and DOB. A random variable could be any one attribute or a collection of attributes of the Students table in the sense that for a randomly chosen tuple, the value of its ID cannot be pre-determined and can only be one of all the possible values for ID.

2.4 Particulars of random events

Furthermore, Xu, Feng and Crowe (2008) point out that even though Dretske's definition is plausible, the role that individual events play in our looking at the information content of a state of affairs was overlooked in Dretske's definition. To amend this, Xu, Feng and Crowe [16] put forward a definition of particulars of a random event as follow:

"Let *s* be a selection process under a set C of conditions, X a random event concerning *s*, X_i an instance of s, X_i is a particular of X if X_i is in a state **Ω**, written **Ω** = state(X_i), and X ∋ **Ω**. "

As in the example above, to select a student record from Students table is a random variable, the record happens to be John's is a random event, and one occurrence of John's record is a particular of the random event.

2.5 'Information Content Inclusion' Relations

The term, 'Information Content Inclusion' Relation, was firstly put forward by Feng in 1998 [5]. It was defined as follows: if the particulars of random event Y are in the information content of the particulars of random event X then we say that 'random event Y is in the information content of random event X', and such a relationship between X and Y is called the 'information content inclusion relation', IIR for short.

In addition to the definition above, Xu, Feng and Crowe [16] clarify four types of IIR and their sources shown in the table below:

Information Inclusion Relation: Information content of X includes Y	**Sources**
X, Y: both database random events	Syntactic relations between data constructs and data values
X: a database random event; Y: a real world random event	Semantic values and information content of data
X: a real world random event; Y: a database random event	Rules and processes of database design and database operations
X,Y: both real world random events	Relations between real world objects, business rules

2.5.1 IIR Rules

Xu, Feng and Crowe [16] identify five inference rules for reasoning about IIR with proofs for the soundness and completeness of the rules. The rules are:

Reflexivity: $Y \subseteq X$, then $X \rightarrow Y$

This rule means that if random event Y is contained in random event X then the information content of X includes Y, which is also denoted IIR(X, Y). The rest of the rules can be interpreted similarly.

Augmentation: $X \rightarrow Y$, then $XZ \rightarrow YZ$

Transitivity: $X \rightarrow Y$, $Y \rightarrow Z$, then $X \rightarrow Z$

Union: $X \rightarrow Y$, $X \rightarrow Z$, then $X \rightarrow YZ$

Decomposition: $X \rightarrow YZ$, then $X \rightarrow Y$, $X \rightarrow Z$

2.5.2 Original IIR

Original IIR are those that are identified by applying IIR definition directly to a variety of sources such as the real world, database systems and domain knowledge and are not those that are derivable by using the inference rules on known IIR. For example, Referential Integrity is a kind of constraints in a relational database, from which, original IIR can be derived.

2.5.3 Differences between Functional

Dependencies and IIR

The inference rules for IIR may look similar to those for Functional Dependencies. But there are several differences between them. Xu, Feng and Crowe [16] give a table shown below, which summaries the differences:

	Functional Dependencies	**IIR**
Objects concerned	Attributes in a relation	Events-member of power set of outcomes of a selection process
Characterization of objects concerned (1)	Both random and certain ones are covered	Random
Characterization of objects concerned(2)	Within a DB	DB and the real world – altogether four types
What is based on	Syntactic Characterization	Syntactic, Semantic, Norms, Business rules…
Veridicality	N/A	The veridicality of event X is a necessary condition for X to be qualified as information being carried

3 IIR Closures

3.1 The Closure of a set of IIR

Let F be a set of IIR. F closure (denoted F^+) is the set of IIR implied by F. $F \subseteq F^+$. If $F= F^+$, F is called a complete set of IIR in the sense that no more IIR can be derived from it by using the IIR rules.

3.2 IIR Closure of a Random Event

All random events that are derivable by using the IIR inference rules on a given set of original IIR and therefore are in the information content of the given random event constitute the IIR closure of the random event. For example, ‘Student ID = B001’ is a random event, and ‘Student Address = 1 High Street’ is in its information content. Likewise, ‘Student Postcode = PA1 2BE’ is in that of ‘Student ID = B001’. Through Transitivity, ‘Student Postcode = PA1 2BE’ is also in the information content of ‘Student ID = B001’. All such random events as ‘Student Postcode = PA1 2BE’ and ‘Student Address = 1 High Street’ would constitute the IIR closure of ‘Student ID = B001’. Let x_1 denote ‘Student ID = B001’, then we use x_1^+ to done the IIR closure of x_1.

3.3 IIR between Attributes and IIR Closure of an Attribute

Let X be an attribute. X can be taken as a random variable, and by taking one of the values that X can possibly take, we may say that X contains a set of random events. In other words, X can be seen as the aggregation of all its random events. As a random event may have another in its information content through having an IIR with it and the latter is contained by another random variable, two random variables may form a relationship between them based on IIR. If every random event of Y is in the information content of at least one random event of X, then we say that attribute Y is in the information content of attribute X, denoted IIR(X, Y). All such attributes as Y that are logically implied by a given set of IIR, which can therefore derived by using the IIR rules, constitute the IIR closure of X denoted X^+. That is, X^+ denotes the set of all attributes such that for everyone of which each of its contained random events has an IIR with at least one of X’s random event, that is, the former is in the *information content* of the latter.

For example, we would have IIR(Student ID, Student Address), which means that Student Address is in the information content of Student ID. By the IIR rules, we can get IIR(Student ID, Student Postcode). Therefore $(\text{Student ID})^+$ would include Student Address and Student Postcode, among others.

3.4 Derivation of three Levels of Closures with Oracle

Our Oracle implementation of IIR reasoning derives three levels of *closure*s, namely, those between classes or tables, those between attributes and those between values of attributes. The first two are closures between random variables, and the third between random events. That is to say, the first two closures are concerned with relationships between random variables. However, IIR is a relationship between random events. The former relationship is similar to IIR, but it needs to be clarified. In the Oracle implementation, IIR closure of class/table X contains all classes or tables that are implied by class/table X and that are inferable by using IIR rules against a given set of IIR. We observe that the rules for random variables are similar to those for IIR, which are applicable only to random events. In this case, the IIR closure of table X is the set of classes/tables implied by X.

Similarly, the IIR closure of an attribute (also a random variable) deals with attributes of a table. For example, variable X's closure contains all attributes implied by variable X. The IIR closure of a random event for a relational database deals with data values. More precisely, a random event in a database in our formulation exists in the form of combinations of attributes and values in database. In other words, an attribute and a value construct a pair that is seen as a random event in a database. As a result, the IIR closure of random event X contains a set of pairs of all attributes and values implied by random event X.

3.5 Why computing IIR closures

To compute F^+ given F, we can compute instead X^+ for all X, which is normally easier than computing F^+ directly. Once X closure is known, to know if IIR(X, Y) holds given F (i.e., whether it is implied by F) is a matter of verifying if Y is in the X closure or not. If so, the IIR holds. Otherwise, as far as the given F goes, the IIR does not exist.

3.6 A Flow Chart of the Basic Concepts

With all the above basic notions in mind, a flow chart can be constructed, which depicts how the basic concepts are linked with one another.

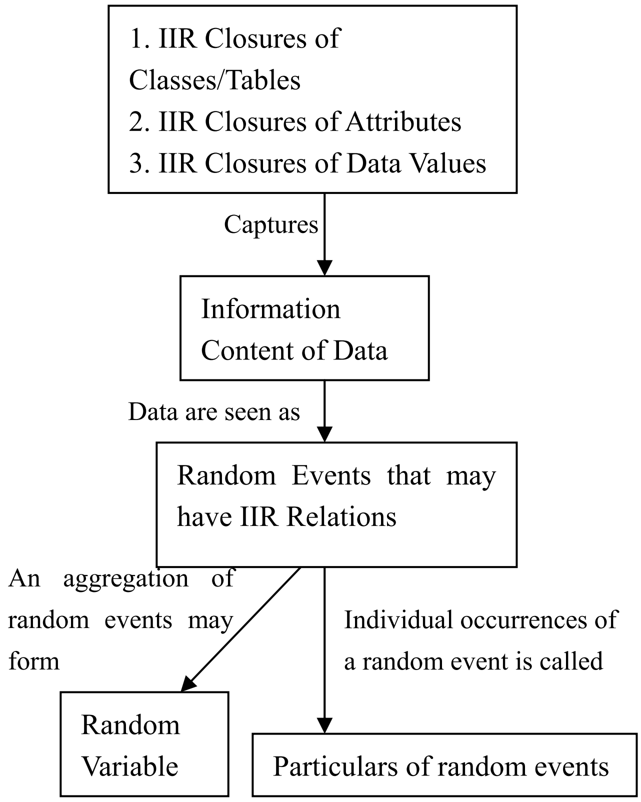

Fig. 1 A Flow Chart of the Basic Concepts

This diagram shows that closures of the three levels capture and formalize the information content of data in a database. Data in a database are now formalized as random events and random variables, and the 'information content inclusion' relation (IIR) can therefore be identified. Thus, the IIR closure of X is the information content of X as far as the data in a database and a given set of identifiable IIR between data go.

4 A System for Reasoning about Information Content of Data in a Database

With the idea of IIR and associated other notions, we have created a system for reasoning about the *information content* of data in a database by taking into account the ideas of Wang and Feng [14], and Eessaar [4], Intuitively, the system works like this. To select a student from a Students table is seen as a random event. And the term 'particular' is used to describe a single occurrence of a random event. For example, student John's record happens to be selected from the Students table, and this particular occurrence of selection of John's record is a 'particular' of the random event that the record happens to be John's. A random variable may be seen as an aggregation of random events. In a table, an attribute can be seen as a random variable because it normally contains many random events in it. For example, Student Name is random variable, which contains Student Name being John and Student Name being Herman, among others. The IIR closure of Student ID being B001, for example, contains Student Name being 'John', Student Major being 'history' and Class Name being 'BD445'. If a user queries about the class name about John, the query can be answered by searching in this IIR closure of Student ID being B001. That is, once IIR closures are known, queries can be posed on these closures. This way some information that cannot be found by conventional queries may be discovered.

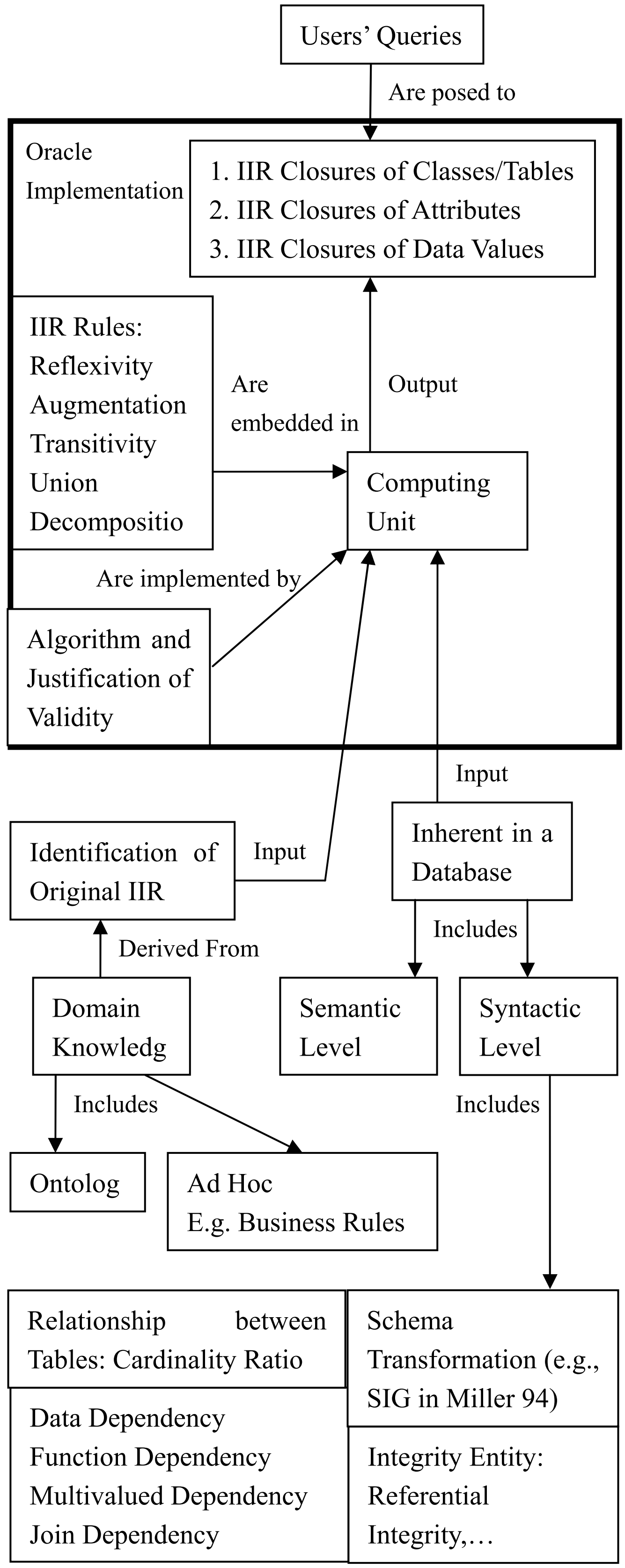

Fig. 2 A System for Reasoning about Information Content of Data in a Database

As depicted in Fig. 2, the system consists of three main parts. The upper part is where users pose queries to the Oracle implementation of the system. The middle part is the Oracle implementation of information content reasoning. The lower part includes a variety of sources of original IIR, mainly from domain knowledge and the syntactic and semantic properties of the database that are inherent to it.

The form of the queries is the conventional SQL. Most programming efforts were put on the implementation of computing the IIR closures. The core algorithm is based on the IIR rules. Original IIR were then added into the unit. This is one of the most difficult parts in the programming as when more original IIR were discovered more computation capability has to be added into the program such that the closures can continually increase accordingly. The output of the unit is simple. Three kinds of closures are provided by the system separately or together depending on the need of the user. Importantly, these outcomes of closures contain information content of random events. User queries, then, be posed on these closures. Thus, more information can be discovered through queries.

The process of discovering original IIR could be extremely hard. There is a variety of sources out there that could potentially contain huge amount of original IIR [13]. The two main sources though are domain knowledge and the properties of the database *per se*. The latter can be further divided into those of semantic and syntactic levels respectively. Hereinto, the syntactic level includes plenty of constraints, which can be directly translated into original IIR such as data dependency, integrity rules and the cardinality ratio between tables.

4.1 Oracle Implementation

The Oracle Implementation of the prototype of the framework was carried in two stages. In the first one, a simplified implementation of the prototype was built to test if the whole ideas can actually work on Oracle DBMS. Then, a more functionally comprehensive implementation was built to tackle example queries. Both of the above two implementations were developed by using PL/SQL [12].

4.1.1 The Preliminary Implementation

In this stage, the IIR rules were embedded in a computing unit of the implementation. These rules constitute the core of an inference engine. The whole process of program development of computing unit had been completed in advance.

For the prototype, we only added some example original IIR into the computing unit. And the outcomes can be seen as an abstract of closures. The following example shows the detail of the implement of the prototype.

We assume that the following IIR are given: F {AB→C, C→A, BC→D, ACD→B, D→EG, BE→C, CG→BD, CE→AG}, in which X→Y is a simplified version of IIR(X, Y), which means that the information content of X includes Y.

Supposing we wanted to know the IIR closures of all combinations of attributes based on the above given IIR. Then, these IIR were imported into the computing unit. The computing unit then computed the IIR closures, which were stored in a table in an Oracle database. Then we used the SELECT command to display the result as follow:

```
XNO XDET                           XCNO XDEP
---- -------------------- ---------- ------------
  45 EDCB                              1 EDCBAG
  46 FEDCB                             0 FEDCB
  47 FEDCB                             1 FEDCBAG
  48 GFEDCB                            0 GFEDCB
  49 GFEDCB                            1 GFEDCBA
  50 DB                                0 DB
  51 DB                                1 DBEG
  52 DB                                2 DBEGC
  53 DB                                3 DBEGCA
  54 EDB                               0 EDB
  55 EDB                               1 EDBCG

XNO XDET                           XCNO XDEP
---- -------------------- ---------- ------------
  56 EDB                               2 EDBCGA
  57 FEDB                              0 FEDB
  58 FEDB                              1 FEDBCG
  59 FEDB                              2 FEDBCGA
  60 GFEDB                             0 GFEDB
  61 GFEDB                             1 GFEDBC
  62 GFEDB                             2 GFEDBCA
  63 EB                                0 EB
  64 EB                                1 EBC
  65 EB                                2 EBCGDA
  66 FEB                               0 FEB

XNO XDET                           XCNO XDEP
---- -------------------- ---------- ------------
  67 FEB                               1 FEBC
  68 FEB                               2 FEBCGDA
  69 GFEB                              0 GFEB
  70 GFEB                              1 GFEBC
  71 GFEB                              2 GFEBCDA
  72 FB                                0 FB
  73 GFB                               0 GFB
  74 GB                                0 GB
  75 C                                 0 C
  76 C                                 1 CA
```

Fig. 3 IIR Closures Computed (a Screen Dump)

In Fig. 3, XNO counts the number of combination of attributes. The combinations *per se* are stored under column name XDET. XCNO records the number of times of a attribute combination been computed. 0 means the first time of computing, 1 means the second time of computing and so forth. For example, attribute combination FEB in NO. 66 gets 0 in XCNO and FEB in XDEP, which means that the closure of FEB is FEB after first computing. Similarly, FEB in NO. 67 gets 1 in XCNO and FEBC in XDEP, which means that the IIR closure of FEB is FEBC after two iterations of the computing.

As shown above, all the closures are shown in the right hand side. For each attribute combination, it has at least one closure that is itself. Some combinations have more than one closure because computing unit detects certain IIR implied by the attributes within the combination. Thus, a new closure was then produced and stored in the table. Then, the process repeated over and over again until the unit cannot detect any related IIR for the attributes. For the programming perspective, the program stops once the attributes in the left hand side of the original IIR do not include any more attribute that is in the information contents of them.

4.1.2 The Comprehensive Implementation

In addition to the core computing unit, the comprehensive implementation of our prototype integrates many more original IIR based on a realistic example.

Suppose the following three tables are stored in an Oracle database.

Students (sid, sname, stmajor, yr, age)

Class (cname, time, room)

Enrolment (sid, cname)

In the tables above, *sid* is the primary key of Students, *cname* is the primary key of Class, *sid* of Enrolment is a foreign key referencing Students and *cname* of Enrolment is a foreign key referencing Class. Sid and *canme* combined as the composite primary key of Enrolment.

The tables were populated with sample records. As Fig. 2 shows, referential integrity is one of the sources from which original IIR are derived. Thus, the constraints above were translated and integrated into the computing unit, in addition to the IIR rules. Suppose that students in different subjects fancy a variety of sports. For example, history students fancy swimming, while geology students like diving. This could be ad hoc business rules of domain knowledge from which original IIR rules can also be derived, and they are integrated into the computing unit.. As a result, the size of IIR closures expanded accordingly. We now give an example below.

Suppose we want to know the IIR closure of 'SID 150', i.e., Student ID being 150. 'SID' and '100' would be inputted into the computing unit. The IIR closure would then be presented on the screen:

```
SQL> START  f:\oracle_scripts_lab\Steal_Me_Again\test_co
Parks geology So 21
BA200 TTH9 SC110
Parks geology So 21
BA200 TTH9 SC110
Favorite sports is diving

PL/SQL procedure successfully completed.
```

```
SQL> select * from newfacts;

       NNO NAT                  NTN
---------- -------------------- ------------
         1 CNAME                ENROLLMENT
         2 SNAME                STUDENTS
         3 STMAJOR              STUDENTS
         4 YR                   STUDENTS
         5 AGE                  STUDENTS
         6 TIME                 CLASS
         7 ROOM                 CLASS

7 rows selected.
```

Fig. 4 An IIR Closure Computed (a Screen Dump)

As Fig. 4 shows, the closure is made up of two parts. One part contains data values 'Parks geology So 21 BA200 TTH9 SC110 diving' etc shown in the top half of Fig.4. The other part shown in the bottom half part of Fig. 4 contains the attributes to which the values belong and the tables to which the attributes belong. Thus, the above closure of 'SID 150' can be read as having CNAME = BA200, SNAME = Park, STMAJOR = geology, YR = So, AGE = 21, TIME = TTH9, ROOM = SC110, and Favourite Sport = diving. The closure above was based on the input of 'SID 150' only. In order words, the information content of 'SID 150' is CNAME BA200, SNAME Park, STMAJOR geology, YR So, AGE 21, TIME TTH9, ROOM SC110, and Favourite Sports being diving. All the information except 'diving' is inferred by using records of the tables stored in the Oracle database, the IIR rules and the original IIR derived from referential integrity. 'Favourite sport is diving' is inferred by using original IIR derived from business rules outside the Oracle database in addition to IIR rules. This example shows that the closure of 'SID 150' contains not only information within the database, but also information from outside the database such as business rules. Once users' queries are posed on the closure, more information will be provided to them.

5 Future Works

With IIR rules, we have discussed the relation regarding information content (i.e., information carrying) between random events. Such a relation at a higher level, i.e., that between random variables is still not clear. How the relations on different levels are connected also deserves further investigation.

By far, the process of identifying original IIR is done manually. However, ideally, the system could identify original IIR automatically depending on the need of user.

More hidden information within database should be discovered with the increase of original IIR derived from database itself and outside sources. Original IIR rules derived from sources like an ontology has not been implemented yet. The programming structure of the computing unit has not been examined in terms of efficiency and robustness. In addition, a graphic interface should be integrated into the programme to help users pose queries directly. Distributed systems may be taken into account as well in the future.

6 Conclusions

In this paper, we have proposed a novel approach to information content reasoning of databases. We gave a set of basic concepts and described a prototype of a system for such purpose. A number of examples were used to test our system. With information sources outside database imported into the system, the information content of a random event (data values) within the database expanded dramatically. Users could make the most of the information content by posing queries. Thus, more information can be discovered than conventional queries. Technically speaking, the increase of random event's closures is based on the boost in original IIR. However, identification of original IIR rules could be extremely hard due to wide range of sources outside database. However, once original IIR have been identified and then integrated into the computing unit of our system, the system provides a powerful engine for users to query a database.

References:

[1] Barwise J and Seligman J (1997), *Information Flow: the Logic of Distributed Systems*, Cambridge University Press, HISBN 0-521-58386-1H

[2] Connolly T and Begg C (2004), *Database systems: a practical approach to design, implementation, and management*, Harlow: Pearson Education.

[3] Dretske F (1999), *Knowledge and the flow of information*, Stanford, Calif.: CSLI Publications.

[4] Eessaar E (2006), Guidelines about usage of the complex data types in a database. *WSEAS Transactions on Information Science and Applications*, v 3, n 4, April, 2006, pp. 712-719.

[5] Feng J (1998), The 'Information Content' Problem of a Conceptual Data Schema, *SYSTEMIST*, Vol.20, No.4, pp.221-233, November 1998. ISSN: 0961-8309.

[6] Feng J and Crowe M (1999), The Notion of 'Classes of a Path' in ER Schemas, *Proceedings of Third East European Conference on Advances in Databases and Information Systems*, ADBIS'99, Springer, Berlin.

[7] Feng Y (1986), *Database Foundation*, Hua Zhong Institute of Technology Press.

[8] Floridi L (2005), Is Semantic Information Meaningful Data? *Philosophy and Phenomenological Research*, 70 (2), pp.351-370

[9] Hull R (1986), Relative information Capacity of Simple Relational Database Schemata, *SIAM Journal of Computing*, 15(3): 856-886

[10] Miller R J, Ioannidis Y E and Ramakrishnan R (1993), The Use of Information Capacity in Schema Integration and translation, *Proc. of the Int'l Conf. on Very Large Data Bases*, Dublin, Ireland: 120-133

[11] Miller R J, Ioannidis Y E and Ramakrishnan R (1994), Schema Equivalence in Heterogeneous Systems, Bridging Theory and Practice, *Information Systems*, 19(1): 3-31

[12] Urman S, Hardman R and McLaughlin M (2004), *Oracle Database 10g PL/SQL Programming*, McGraw-Hill/Osborne.

[13] Wang S, Lin C and Feng J (2007), A Hermeneutic Approach to the Notion of Information in IS, *Proceedings of the 7th WSEAS international conference on simulation, modeling and optimization*, 2007, pp.400-404.

[14] Wang Y and Feng J (2007), FCA Assisted IF Channel Construction towards Formulating Conceptual Data Modeling, *WSEAS Transactions on Systems*, v 6, n 6, June, 2007, pp.1159-1167.

[15] Xu H and Feng J (2002), Towards a Definition of the 'Information Bearing Capability' of a Conceptual Data Schema, *Systems Theory and Practice in the Knowledge Age*, (E. Ragsdell et al.), Kluwer Academic/Plenum Publishers, New York, ISBN 0-306-47247-3

[16] Xu K, Feng J and Crowe M (2008), Defining the Notion of 'Information Content' and Reasoning About it in a Database, *Knowledge and Information Systems*, Springer-Verlag London Limited. DOI 10.1007/s10115-008-0129-3