THE CRUD CUBE

Alberto Sulaiman [1,2], Jano Moreira de Souza [1,3] , Julia Celia Mercedes Strauch [1,4]
[1]*Coordenação dos Programas de Pós-graduação em Engenharia, Universidade Federal do Rio de Janeiro - COPPE/UFRJ; Cidade Universitária, Centro de Tecnologia, bloco H, sala 319, Ilha do Fundão Caixa Postal: 68513 - Rio de Janeiro (RJ) - Brasil*
[2]*Departamento de Mercado Aberto, Banco Central do Brasil - DEMAB/BACEN*
[3]*Instituto de Matemática, Universidade Federal do Rio de Janeiro - UFRJ - Rio de Janeiro - Brasil*
[4]*Escola Nacional de Ciências Estatísticas, Instituto Brasileiro de Geografia e Estatística - ENCE/IBGE*

Abstract

This work introduces a design pattern, a fact table, for OLAP and Data Mining use, named CRUD Cube. The CRUD Cube is an abstraction idealized from the CRUD Matrix concept extended by an extra dimension: the time dimension. Differently from the typical use of the CRUD Matrix, the Business System Planning, the example application presented in this paper generates workflows and business rules from database endogenous logs.
Keywords: Data Warehousing, OLAP, Data Mining, Design Pattern.

1 Introduction

This paper is a result obtained from [1, 2] and introduces a pattern for a fact table named CRUD Cube. The CRUD Cube is an abstraction idealized from the CRUD Matrix concept [3] extended by an extra dimension: the time dimension.
Section 2 formalizes the CRUD Cube concept and specifies the target problem; section 3 outlines CRUD Cube and Business Rules; section 4 shows examples of Business Rules obtained from a CRUD Cube; section 5 outlines CRUD Cube

and Workflows; section 6 shows examples of Workflows obtained from a CRUD Cube. Section 7 concludes the paper.

2 Formal Model

Let $T = t_1, t_2, ..., t_m$ be the set of transactions
Let $A = a_1, a_2, ..., a_n$ be the set of files/tables of the same application; and
Let $H = h_1, h_2, ..., h_p$ be the set of time periods observed, where $h_n < h_{n+1}$; and
Let the relation $R \subseteq A \times T \times H = t_1a_1h_1, t_1a_2h_1, ..., t_ma_nh_p$ be the relation that represents the table of facts CRUD Cube, where each $t_ia_jh_k$ is an operation.
C = *Create* /* Transaction t_i creates a record in A_j in the time H_k*/
R = *Retrieve* /* Transaction t_i retrieves a record in A_j in the time H_k*/
U = *Update* /* Transaction t_i updates a record in A_j in the time H_k*/
D = *Delete* /* Transaction t_i deletes a record in A_j in the time H_k*/

	A1	**A2**	**A3**
T1	RU		C
T2	R	U	R
T3	C	CUD	U

H_1	**A1**	**A2**	**A3**	**H2**	**A1**	**A2**	**A3**	**H3**	**A1**	**A2**	**A3**
T1	R			T1	U			T1			C
T2	R			T2		U		T2			R
T3	C			T3		C		T3		U	

H_4	**A1**	**A2**	**A3**	**H5**	**A1**	**A2**	**A3**	**H6**	**A1**	**A2**	**A3**
T1				T1				T1			
T2				T2				T2			
T3		D		T3			U	T3			

Figure 1: Example of a CRUD matrix with decomposition into a CRUD Cube

Taking into consideration Figure 1 it is possible to imagine the following OLAP queries, shown on Figure 2:

- Case 1: fixing the transaction and the file, varying the dimension time, it is possible to determine CRUD's commands order of execution to a certain file, which is referenced by the "Y" transaction. Exemplifying: let the transaction and the target file, T3 and A2 of Figure 3 respectively, we know that the operations executed are C, U, and D, in this order;

- Case 2: fixing only the transaction and varying the dimensions file and time, it is possible to determine the execution order of the commands (CRUD) for each file that is referenced by the transaction "X". Exemplifying: let the target transaction, T3 of Figure 3, we know that the operations executed are C(A1), C(A2), U(A2), D(A2), U(A3), *commit*, in this order ;
- Case 3: fixing a range (period) of time it is possible to determine which transactions occur in that period, in a way that in the future it be possible to group transactions that occur closely. As an example, for times varying from H1 to H5: T_1: R(A1), U(A1), C(A3), *commit*; T_2: R(A1), U(A2),R(A3), *commit*; T_3: C(A1),C(A2),U(A2),D(A2),U(A3), *commit*; in this order.

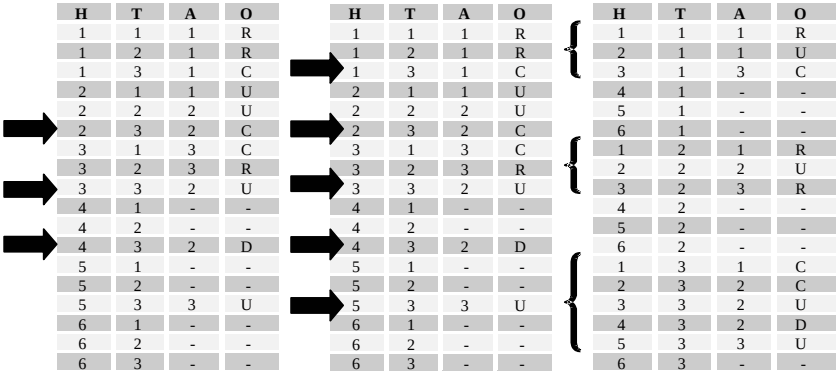

Figure 2: Cases 1, 2 and 3

And specially, having obtained a CRUD Cube from the endogenous log, it is possible to suppress the time dimension, leaving only the dimensions transaction and file/table, as well as its operations, with no repetitions, obtaining the CRUD Matrix, by reverse engineering, as was proposed. This is shown in Figure 3.

2.1 The Target Problem

In this work, the CRUD Cube fact table is introduced. For example, when a roll-up operation, an OLAP operation, against a CRUD Cube is applied, a CRUD Matrix is generated automatically. Besides this, a CRUD Cube can also be mined using at least two techniques: 1. Associative Rule Mining; and Mining of Sequential Patterns.

In the first case, the result permits the attainment of a couple of business rules. In the second case, the results permit the attainment of parts of workflows.

In the specific case of logs coming from the stock auction of the Central Bank of Brazil (OFPUB), in the DMSII/UNISYS environment, two endogenous files of similar format are generated, but of complimentary functions:

1. The first file, of a smaller size, maps all of the programs referenced in a certain auction; and

2. The second file, of a larger size, has the register of all the updates, inclusions and deletions of each data object belonging to any referenced data structure.

Each logical record of the larger file also contains an identifier of the program, which generated it, whose complete name is found in the smaller file.
The extraction of the CRUD Cube happens over the two files and also on the information coming from the catalog. In the experiment exposed in this article the auctions analyzed are those occurred on October 11, 15, 16, 17 and 18 of 2001.

H	T	A	O
1	1	1	R
1	2	1	R
1	3	1	C
2	1	1	U
2	2	2	U
2	3	2	C
3	1	3	C
3	2	3	R
3	3	2	U
4	1	-	-
4	2	-	-
4	3	2	D
5	1	-	-
5	2	-	-
5	3	3	U
6	1	-	-
6	2	-	-
6	3	-	-

	A 1	A 2	A 3
T 1	R U		C
T 2	R	U	R
T 3	C	C U D	U

Figure 3: Generation of a CRUD Matrix

3 The CRUD Cube and the Business Rules

When mining the database's transaction log it is possible to discover rules relative to the functional and control axles of abstraction. These rules can be combined with rules obtained from the metadata to execute the reverse engineering of the business rules with a format similar to the associative rules.
But these associative rules associate CRUD operations in time, in a way that the order in which these operations occur in a database transaction, in this case, is important, as was described in the cases 1, 2, and 3 of the prior section. In [4] the shopping items' order in a transaction, in the mining of associative rules, is irrelevant.
Below, examples of proposals of the functionalities of data mining desirable in the query processing of a DBMS [4], in exogenous applications, for the case of retailing (*market basket analysis*), with its corresponding endogenous [1] for the case of the mining of logs:
Case 1:

- Find all rules that have "Diet Coke" as consequent. These rules may help plan what the store should do to boost the sale of Diet Coke [4];
- Find all of the rules that have "delete record from table A" as a consequent. These queries can help in the identification of business rules of the "inference" or "stimulus/answer kind, whose consequent is pre-determined, or all the types of stimulus which can lead to this answer [1].

Case 2:

- Find all rules that have “bagels" in the antecedent. These rules may help determine what products may be impacted if the store discontinues selling bagels [4];
- Find all of the rules that have “insert a record in table B” as an antecedent. These queries can help in the impact determination of this operation on other transactions, identifying the business rules whose consequents correspond to the predetermined antecedents [1].

Case 3:

- Find all rules that have “sausage" in the antecedent and “mustard" in the consequent. This query can be phrased alternatively as a request for the additional items that have to be sold together with sausage in order to make it highly likely that mustard will also be sold [4];
- Find all of the rules that have “insert record in table C” as an antecedent AND “exclude record from table D” as a consequent. These queries can help in the prospection of certain similar behavioral standards in distinct transactions [1], or more specific inference rules.

Case 4:

- Find all the rules relating items located on shelves A and B in the store. These rules may help shelf planning by determining if the sale of items on shelf A is related to the sale of items on shelf B [4];
- Find all of the rules that relate the operations between the tables X and Y on schema Z. These queries can help in the Information Systems Planning, by the determinations of the existing relations between the transaction Q on table X and the transaction W on table Y [1]. This is an inter-transaction approach, coming close to the treatment of workflows.

4 Examples of attained Business Rules

For the generation of business rules, WIZRULE 3.05, an automatic associative rule generator, was used against the CRUD Cube, with the following parameters:

1. Total number of records: 7242;
2. Minimum probability of rules “if-then”: 0,95 (trust factor);
3. Minimum number of cases: 40 (minimum support of any rule obtained: 0.55%)

The times were grouped in three periods: the first one goes from the early morning until noon, when the supply cadasters are done. The second between noon and 1pm, when the auction actually happens. The third happens after 13 hours when the poll is done. For this specific query eighteen rules were generated, rules from which the following were chosen:

Rule 13: If PGM is 17.00 and CRUD is D
Then PERIODO is 1.00
Rule's probability: 1.000
The rule exists in 496 records.
Significance Level: Error probability < 0.001

Meaning: this rule reports that when program 17 , which is “PROPSTM” executes removal operations, these operations occur in the supply cadasters’ period.
Rule 18: If PGM is 17.00 and DATASTR is 7.00 and PERIOD is 3.00
Then CRUD is C
Rule's probability: 1.000
The rule exists in 224 records.
Significance Level: Error probability is almost 0
Meaning: when program 17 operates over data structure 17 in the poll period, this is a creation operation.
Rule 16: If DATESTR is 26.00 and CRUD is C
Then PERIOD is 2.00 Rule's probability: 0.994
The rule exists in 804 records.
Significance Level: Error probability is almost 0
Deviations (records' serial numbers):
5669, 5672, 5673, 5681, 5682
Meaning: the data structure 26 is created during the auction.
Rule 17: If DATESTR is 26.00 and CRUD is D
Then PERIOD is 3.00
Rule's probability: 1.000
The rule exists in 1198 records.
Significance Level: Error probability is almost 0
Meaning: data structure 26 is eliminated in the subsequent poll period.

5 THE CRUD Cube and the Workflow

In general, transaction logs grouped in time refer to one activity. Sometimes grouped transactions solve part of a larger activity previously defined. In this paper we used the term “abstract transaction” in opposition to “concrete abstraction”, in the following manner:

- Concrete transaction – transaction obtained by reverse engineering by the simple registration of the activities that happened in a period of time; and
- Abstract transaction – registration of the transaction from reverse engineering by obtaining of maximal sequences [5] of activities that represent “a computational interpretation of the business activity” [1], or the computational portion of an activity of the business, implemented in a database.

The theory of Sequential Patterns Mining is described in [5] and, in this work an instantiation for the case of database transactions is done. Evidently what we wish to obtain in a semi-automatic manner are workflows as authentic as possible to the real model, having as inputs the sequences of programs which actually happened in a period of time. The restrictions to the model are the following: 1. A data sequence, or just sequence, is an ordered list of programs; 2. A sequence is a sequential pattern; 3. The order of the programs in each sequence refers to the order of execution of each program; 4. Each program is mapped to an integer identifier; 5. An auction supports a sequence if is contained

in the sequence of programs of this auction; 6. The support of a sequence is a fraction of the total amount of auctions that support this sequence.

6 Examples of parts of Workflows obtained

In the obtaining of the workflow parts, a program was developed based on the theory of [5]. An exploratory study was done on auctions which occurred on October 11, 15, 16, 17 and 18, had the following configurations, for each auction in this case (concrete transactions):

1. Size: 22 - 2 4 7 17 12 16 15 11 18 19 15 5 8 5 4 4 4 8 5 4 3 5
2. Size: 22 - 2 4 8 8 5 4 8 5 5 9 5 4 5 7 17 12 16 15 11 18 19 15
3. Size: 22 - 2 9 4 3 5 3 9 1 8 7 17 12 16 15 11 18 19 15 1 6 10 9
4. Size: 18 - 2 4 7 17 12 16 15 11 18 19 15 10 5 5 5 5 13 14
5. Size: 22 - 2 4 5 7 17 12 16 15 17 12 16 15 11 18 19 15 11 18 19 15 10 4

The maximals obtained were the following (abstract transactions):

1. 2 4 7 17 12 16 15 11 18 19 15 4 Frequency: 2 Size: 12
2. 2 4 7 17 12 16 15 11 18 19 15 5 5 5 5 Frequency: 2 Size: 15
3. 2 4 5 8 7 17 12 16 15 11 18 19 15 Frequency: 2 Size: 13
4. 2 4 5 9 7 17 12 16 15 11 18 19 15 Frequency: 2 Size: 13
5. 2 9 4 5 7 17 12 16 15 11 18 19 15 Frequency: 2 Size: 13
6. 2 4 5 7 17 12 16 15 11 18 19 15 10 Frequency: 2 Size: 13

A more detailed analysis over these partial results can be done on the “strength” that each part of the sequence, in the following manner: for the representative sequences of the five auctions above were obtained the six following maximal sequences. These sequences constitute the abstract transactions which combined, are candidates to parts of the workflow. The sequence “7 17 12 16 15 11 18 19 15” is the strongest: it occurs in 100% of the cases in a contiguous manner, on the sequences of concrete transactions and on the abstract transactions. As the stronger signs constitute parts of the sequence candidates to the dorsal spine of the workflow, this is the case of this sequence.

The sequence “2 4 7” is also strong, but this is a sequence that doesn’t present itself in a contiguous form in all of the sequences of concrete transactions and that suggests, in the abstract transactions, the occurrence of decision structures which can present occurrences of “2 4 5 8 7”, “2 4 5 9 7”,”2 9 4 5 7” and “2 4 5 7”.

The sequence “15 5 5 5 5” suggests the occurrence of a repetition structure, but it would deserve a more detailed analysis for it is still a very weak sign. Figure 6. illustrates a schematic representation of these sequences’ strength, noted on the arrows in bold, in order to represent the frequency in which these abstract transactions occur. The thicker arrows represent stronger sequences, or more frequent ones, while the more narrow arrows represent weaker sequences, or the less frequent ones.

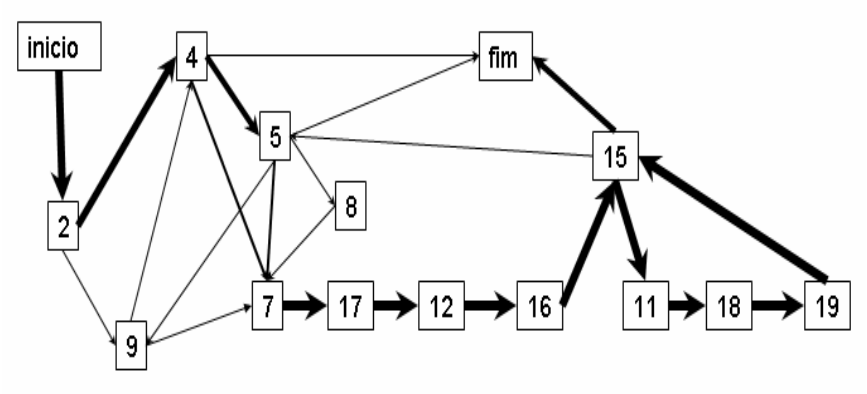

Figure 4: Parts of workflow sequences

7 Conclusion

This paper introduces a pattern for a fact table named CRUD Cube. The CRUD Cube is an abstraction idealized from the CRUD Matrix concept extended by an extra dimension: the time dimension. The CRUD Cube concept had been used to reverse engineer workflows and business rules from database endogenous logs.
The CRUD Cube is a complimentary approach to the reverse engineering solutions to legacy systems is the database log analysis combined with the catalogs of legacy systems.
This work proposes an original approach for the treatment of the reverse engineering of legacy systems: a method that uses data warehouse and data mining techniques in the partial obtaining of business rules and workflows.
An original method for the reverse engineering of business rules and workflows of legacy systems was created based on the non-conventional approach to the data mining of endogenous data. Data warehouse and data mining techniques for the partial extraction of business rules and workflows over these bases were explored. For this, the CRUD Cube concept was created, as well as the possibility of doing OLAP and data mining operations against this fact base. But this approach doesn't solve the problem completely, it thus configures an inexpensive, dependable method of a semi-automatic character.

7.1 Future Perspectives

From the experience of this work it is possible to foresee at least three possibilities:

1. The first and main one is the construction of a methodology which composition contemplates the data mining of endogenous data, combined with the mining of business rules in source programs of the same systems;
2. The second contemplates the last stage of the KDD process, "consolidation of the obtained knowledge" which constitutes a posterior stage to data mining, and which is foreseen as future work; and
3. The third contemplates the log research in distributed and federated environments, which can make the model more complex, although more complete, bringing subjects which would involve not only the temporal dimension, explored in this work, as also the spatio-temporal dimension, referring to the logs' analysis of distributed and federated systems, which bring parallelism and conflict situations not yet solved in the literature.

References

[1] Sulaiman, A. & Souza, J. M., A Decision Support System that Reverse Engineers Abstract Database Transactions - The Conceptual Model. *Data Mining, v. 1, International Conference on Data Mining*, ed . F. F. Ebecken, WIT Press: Boston Southampton, pp. 401-411, 1998.
[2] Sulaiman, A., Data Mining of Endogenous Data, PHD thesis, (in Portuguese) Coppe/UFRJ, 2002.
[3] IBM, *Business System Planning*. Information Systems Planning Guide, Armonk, New York, Publication No. GE 20-0527-4, 1978.
[4] Agrawal, R., Imielinski, T., Swami, A., Mining Associations between Sets of Items in Massive Databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington D.C., pages 207-216, May 1993.
[5] Agrawal, R., Srikant R., Mining Sequential Patterns. *Proceedings of the International Conference on Data Engineering*, Taipei, Taiwan, March 1995.