Data Warehouse Methodology: A Process Driven Approach

Claus Kaldeich and Jorge Oliveira e Sá

Universidade do Minho, Escola de Engenharia
Departamento de Sistemas de Informação, Campus de Azurém
4800-058 Guimarães, Portugal
{cka,jos}@dsi.uminho.pt

Abstract. The current methods of the development and implementation of a Data Warehouse don't consider the integration with the organizational-processes and their respective data. In addition to these current methods, based on demand-driven, data-driven and goal-driven, we will introduce in this paper a new approach to DW development and implementation. This proposal will be based on the integration of organizational processes and their data, denote by: Integrated-Process-Driven (IPD. The principles of this approach are founded on the relation-ships between business-processes and Entity-Relationship-Models (ERM), the Relational Database (RDB) data models. These relationships are originated in the Architecture of Integrated Information Systems (ARIS) methodology. IPD will use the information extracted from the data-driven, on the one side, to match (or define) the AS-IS business processes model. On the other hand, IPD will use the information returned from the demand-driven (required by the DW users) to define the TO-BE business process model based also on the AS-IS model. IPD will integrate the new data models, originated in the TO-BE business processes model, with the DW requirements. The aim of IPD is to define (or to redefine) the organizational processes which will supply the DW with data. The added-value of this approach will be the integration of the previous methods (demand-driven and data-driven) with organizational processes that will treat these sets of informations to be used by the DW. Our approach is also a trigger for business processes reengineering and optimization. Finally, the goal-driven will verify if the IPD achieves the business goals.

1 Introduction

Data Warehouse (DW) systems became an essential component of decision support systems in organizations. DW systems offer access to integrated and historic data from heterogeneous sources to support managers in their planning and decision-making activities. The DW does not create value to an organization; value comes from the use of his data and, of course, the improvement of decision-making activity is the result from the existence of better information available in the DW. The greatest potential benefits of the DW occur when it is used to redesign business processes and to support strategic business objectives [21], [10], but these are also the most difficult benefits to achieve, due to the amount of top management support, commitment, and involvement and the amount of organizational change required.

A. Persson and J. Stirna (Eds.): CAiSE 2004, LNCS 3084, pp. 536–549, 2004.

Building a DW is a very challenging issue and once compared to software engineering it reveals quite a young discipline that does not yet offer well-established strategies and techniques for the development process. Current DW development methods can fall within three basic groups: data-driven, goal-driven and demand-driven. The current methods of the development and implementation of a DW don't consider the organizational processes integration with their respective data. In this paper we will introduce a new approach to DW development and implementation. This proposal will be based on the integration of organizational processes and their data: Integrated-Process-Driven (IPD). IPD will use the information requirements from the analysis of the operational (corporate) data model (ERM) [3] and relevant transactions – the data-driven approach, on one side, to match (or define) the AS-IS business process model. On other hand, IPD will use the information requirements from the end user requirements – the demand-driven approach to define the TO-BE business process model based also on the AS-IS model. IPD will integrate the new data models, coming from the TO-BE business process model, with the DW requirements. The aim of the IPD, is to define (or redefine) the organizational processes which will supply the DW data.

In section 2, three approaches to DW development methods are discussed: data-driven, goal-driven and demand-driven. In section 3, the IPD approach is described. In section 4, the relation between processes, functions and data, based on ARIS are presented and discussed. In section 5, a simple example is showed. This paper concludes with section 6, which presents our final comments and future research.

2 Three Approaches to DW Development Methods

Although it seems to be obvious that matching information requirements of future DW users with available information supply is the central issue of DW development, only few approaches seem to address this issue specifically. Based on whether information demand or information supply is guiding the matching process, demand-driven approaches and data-driven approaches can be differentiated. A special type of demand-driven approach is to derive information requirements by analyzing business processes by increasing detail and transforming relevant data structures of business processes into data structures of the DW. This approach is named goal-driven. All three approaches are described in detail:

- Data-Driven (or supply-driven) approach: the DW development strategy is based on the analysis of organisational data models and relevant transactions [9]; this is completely different from the development of classical systems, which have a requirement-driven development life cycle. The requirements are the last thing to be considered in the decision support development life cycle; they are understood after the DW has been populated with data and results of queries have been analysed by users. This approach ignores the needs of DW users a priori. Organizational goals and user requirements are not reflected at all [7], [8]. However, this approach risks to waste resources by handling many unneeded information structures. Moreover, it may not be possible to motivate end users sufficiently to participate in the project once they are not used to work with large data models developed for and by specialists [4].

- Goal-Driven approach: the first stage of this approach is the process's derivation which determines the goals and services the organization provides to its customers. Then, the business process is analysed to highlight the customer's relations and their transactions with the process under study. In a third step, sequences of transactions are transformed into sequences of existing dependencies that refer to information systems. The last step identifies measures and dimensions needed to design the DW [1], [16]. For decision processes, however, a detailed business process analysis is not feasible because the respective tasks are often unique and unstructured or, what is even more important, because decision makers/knowledge workers often refuse to disclose their processes in detail.
- Demand-Driven (or user-driven) approach: this approach assumes that the organization goal is the same for everyone and the entire organization will therefore be pursuing the same direction. It is proposed to set up a first prototype based on the needs of the business. Business people define goals and gather, prioritise as well as define business questions supporting these goals. Afterwards the business questions are prioritised and the most important business questions are defined, to identify data elements terms, including the definition of hierarchies [22].

These approaches are aimed to determine information requirements of DW users. End users alone are able to define the business goals of the DW systems correctly. So, end users should be able to specify information requirements by themselves. However, end users are not capable to specify objectively unsatisfied information requirements, once: their view is subjective by definition, they cannot have sufficient knowledge of all available information sources, and they only use a specific business unit's interpretation of data. Moreover, end users can often not imagine which new information the DW system could supply [2], [4].

To minimize this it is possible to use a catalogue for conducting user interviews in order to collect end user requirements, or by interviewing different user groups in order to get a complete understanding of the business [18].

As described above all approaches have positive and negative aspects, but our objective is to merge "all" positive aspects to a new approach - IPD – Integration Process Driven.

3 IPD Approach

This approach will be based on the integration of organizational processes: Integrated-Process-Driven (IPD). The principles of this approach are based on the relationships between organizational-processes and Entity-Relationship-Models (ERM) (data models) see figure 1. These relationships come from the Architecture of Integrated Information Systems (ARIS) [19], [20].

IPD will use the information extracted from the data-driven, on one side, to match (or define) the AS-IS organizational process model. On the other hand, IPD will use the information returned from the demand-driven (required by the DW users) to define the TO-BE organizational process model based also on the AS-IS model. IPD will integrate the new data models, originated in the TO-BE organizational process model, with the DW requirements. The aim of the IPD, is to define (or to redefine) the organizational processes which will supply the DW data. The added-value of this

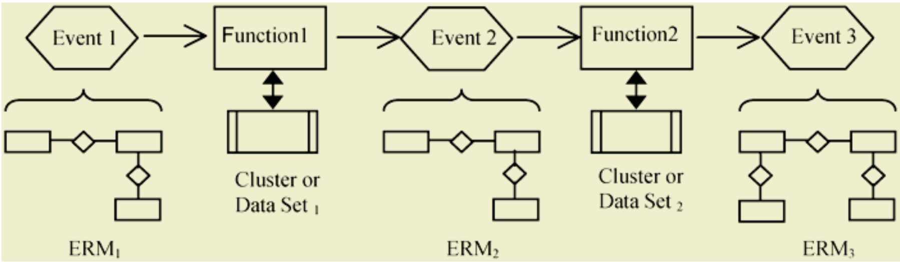

Fig. 1. Event (driven) – Process Chain

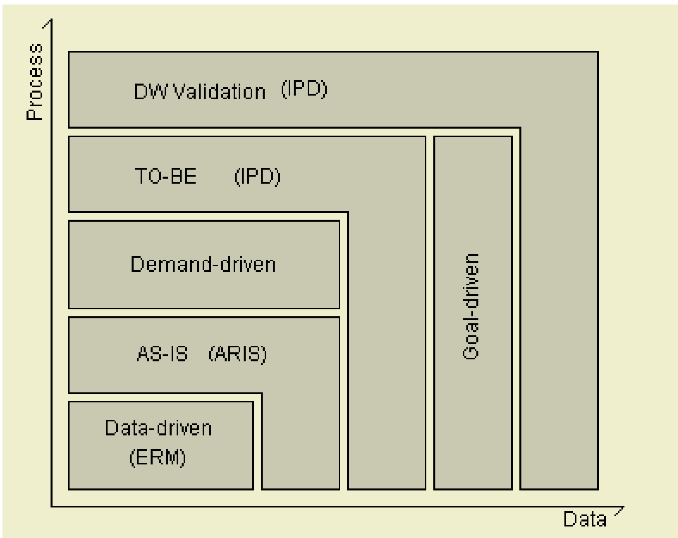

Fig. 2. IPD model

approach will be the integration of the previous methods (demand-driven and data-driven) with organizational processes that will deal with these sets of information's to be used by the DW. Our approach is also a trigger for organizational processes optimization. Finally, the goal-driven will verify if the IPD achieve the business goals, see figure 2.

The relationship between organizational-processes and the respective data sets are trivial. But, not so trivial are the relationships between 'combinations' or 'transformations' from data into sequences of processes (later we will determine this 'combinations' or 'transformations' by *congruencies* of data [13]). These sequences of processes can be parallel, synchronous, asynchronous and so on. Data can be 'derived' from a data set 'transformed' by a process or process-sequences. Data can be the result of *congruencies* of data extracted from different data sets coming from different sources through complex sequences of processes.

In this sense, it is easy to see that a DW can be defined, developed and implemented by different ways to achieve several goals.

Whenever we talk about *data-integration* or *process-integration* along with organizational-processes must be considered the *integration* defined by the *Enterprise Resource Planning* (: ERP) (as an Integrated Information System).

Different grades of data-integration can be achieved in an ERP. For example, the printout of an invoice in an ERP can generate data only for:

1. the Sales-Department: data for the update of accumulated invoice amount/client or accumulated invoice amount /period) or for,
2. the Accounting-Department: data for the update of valued-added-tax accounting/period.

But, the grade of data-integration can be higher and the printout of an invoice in an ERP can generate data also for:

3. treasury: all necessary (direct or derived) data for a Cash-Flow-Simulation until a date-line, and
4. Decision-Support-System (DSS): up to all necessary data for the update of some micro-economics indexes, like a profit-function of a single product/set of products and so on.

4 Organizational Processes Modeling

Concerning to Organizational-Processes-Modelling (OPM), we will use the ARIS[1] regarding important aspects of *integration*. The aim of the modelling with ARIS will define the relationships between functions (as an indivisible element of a process) and respective data [19].

Remarkable is the fact that, depending on the grade of data-integration in an Integrated Information Systems, e.g. an ERP can have multiples processes-chains (interactive, automatics or batch) to increase the data-set, beginning on 'basic' data (like the data from a new invoice) until derived-data (like accumulated invoice amounts up until to the cash ratio) [20].

As well important as the multiples processes-chains is the fact that an ERP can have over thousands of processes-chains and thousands upon thousands transactions which access a Database System to create, update or delete data. Basically, all these data are coming from the organizational-processes and will feed the DW and the DSS.

To support the IPD scope it will be necessary to define some algebraic structures. These structures are connected to the definition of Congruence and Tolerance Relations for Relational Models (in sense of Database Systems) [11],[12], [13], [14]. The aim of these definitions is to apply some algebraic formalism to describe the integration between organizational-processes and data models (ERM). These relations will be used, also, to justify by formalism in the transition from the AS-IS to TO-BE organizational-processes models. The IPD will underline the integration between organizational-processes, data models (ERM).

---

1 © IDS-Scheer, Saarbrücken, Germany.

4.1 Definition: Relational Database (RDB)

A Relational Database, *RDB := (FL, I, IC)*, is defined as:

1. *FL := (S, W)* is a formal language, where:
   1.1. *S* is a set of symbols, and
   1.2. *W* is a set of words defined by elements of *S*.
2. *I* is a *interpretation* of *FL.*
3. *IC* is a set of formulas of *FL*, with will define the Integrity Constraints of the *RDB*: $IC := \{\delta_i \mid \delta_i : \forall \varepsilon_j \rightarrow \psi_k;\ i,j,k \in \{1,\ldots,n\},\ j \neq k;\ \delta,\varepsilon,\psi \in W\}$ ■

4.2 Definition: The Relation *R* of a *RDB*

A relation $R := (Sch_R, D_R, FD_R, T_R)$, is defines as:

1. $Sch_R := \{at_1, \ldots, at_n\}$, is a set of attributes of *R*.
2. D_R is the set of Domains of the dos attributes of *R*:
   $D_R \coloneqq D_{at_1} \cup \ldots \cup D_{at_n}$.
3. FD_R is the set of functional dependencies of *R*:
   $FD_R \subseteq \{F_1 \rightarrow F_2 \mid F_1, F_2 \subseteq Sch_R;\ F_1 \neq F_2\}$. ■

4.3 Definition: *Function-Data* Relation *(fdr)*

Given a function f_i, which is defined as a set of instructions[2] that process a data set. The *function-data* relation is defined as:

1. Let be the structure: $fdr' := (f_i, \{d_{i_1}, \ldots, d_{i_n}\})$
   1.1. Where f_i is a function, and
   1.2. the set of data processed by f_i is: $\{d_{i_1}, \ldots, d_{i_n}\}$
2. Applying the Decomposition Rule[3] on *fdr'* :
   $f_i \rightarrow \{d_{i_1}, \ldots, d_{i_n}\}) \Rightarrow f_i \rightarrow d_{i_1}, f_i \rightarrow d_{i_2}, \ldots, f_i \rightarrow d_{i_n}$.
3. The *function-data* relation for the function f_i, is defined as:
   $fdr_{f_i} := (f_i \rightarrow d_{i_k}),\ k \in \{1,\ldots,n\}$.
4. In this sense is easy to define the *function-data classes* of a family of functions, will be: $fdr_{f_{i,n}} := \bigcup_{k=i}^{n} fdr_{f_k}$.

■

[2] Concern about instructions of a formal language, for example: C++.

[3] Analogous to the *decomposition rule* of the Relational Theory, applied to the Functional Dependencies.

Based on the definitions 0 and 0, some questions emerge:

1. Which following relations can be defined based on these relations to achieve the proposed goal?
2. How these above mentioned new relations complement the definitions 0 and 0?

Strictly speaking these questions derive from some simples ideas:

The definition of FD_R are included in a definition of a relation R (definition 0), which describe the set of functional dependencies of R, where a set of data-attributes, represented by F_1 implies an other set of data-attributes, namely F_2 : $F_1 \rightarrow F_2$; why not to define a relation based on the relation fdr (definition 0) to link functions (obtained from organizational-processes) through the related data to a further extended set of data (describe by the Entity-Relationship-Models)? (Further can considered other relationships between data-attributes or data.)

The AS-IS model, represented by a set of EPC's, will define the executions orders of the functions into a process (and processes sequences). The over mentioned functions orders will define the order of the respectively data processing.

Each element of the demand-driven data set can be defined as a semantic conclusion from the data-driven data set (also denoted by 'basic data') and an additional data set coming from the TO-BE organizational-processes model and integrated by the IPD.

Integrated by the IPD, for the demand-driven data set can be defined:

i. The set of functions which will process these data, they executions orders (based on the AS-IS model and the TO-BE model (represented trough new EPC's)).

ii. The matching of all processes and respective data with the goal-driven: the validation of the TO-BE model generated by the IPD to support the DW-model.

4.4 Definition: *Dependency-Data* Relation *(ddr)*

The *ddr* is defined as:

1. Given FD_R , like in the definition 0 .
2. Give $F_1 := \{at_1,..., at_j\}$ and $F_2 := \{at_{j+1} ,..., at_n\}$ then:
3. $F_1 \rightarrow F_2 \approx (at_1 \wedge ... \wedge at_j) \rightarrow (at_{j+1} \wedge ... \wedge at_n) \approx \{(at_1 \rightarrow at_{j+1}), (at_1 \rightarrow at_{j+2}),..., (at_1 \rightarrow at_n), (at_2 \rightarrow at_{j+1}), (at_2 \rightarrow at_{j+2}),..., (at_2 \rightarrow at_n), ... \}$.
   Expressed as binary relation:
   $\{(at_1, at_{j+1}), (at_1, at_{j+2}),..., (at_1, at_n), (at_2, at_{j+1}), (at_2, at_{j+2}),..., (at_2, at_n), ...\}$.
4. Now, let $e_j \in Dat_i$, $j \in \{1,…,m\}$, $i \in \{1,…,n\}$; be the extensions of the attributes $at_1 ,..., at_n$. Let I be a Interpretation of $F_1 \rightarrow F_2$, then:
   $I(F_1 \rightarrow F_2) \subseteq \{ \{(e_1 , e_{j+1}), (e_1 , e_{j+2}),..., (e_1 , e_n), (e_2 , e_{j+1}), (e_2 , e_{j+2}),..., (e_2 , e_n), ...\}$.
   In this way the relation *rdd* is defined as:

$ddr \subseteq I(F_1 \rightarrow F_2) \subseteq \{(e_1, e_{j+1}), (e_1, e_{j+2}),..., (e_1, e_n), (e_2, e_{j+1}), (e_2, e_{j+2}),..., (e_2, e_n), ...\}$.

■

Based on the definitions above, is possible increase the semantics of the *integration* concept.

4.5 Definition: *Auxiliary* Relation *(auxr)*

Given the function-data relation: $fdrf_i := (f_i, d_j),..., (f_i, d_n))$ and dependency-data relation: $ddr := \{(e_1, e_k), (e_1, e_{k+1}),..., (e_1, e_n),(e_2, e_k), (e_2, e_{k+1}),...,(e_2, e_n), ...\}$.

The *auxr* for the function f_i will be defined as follow:

1. $auxrf_i := \{(f_i, e_k) \mid \forall (f_i, d_j) \forall (e_m, e_k) : d_j = e_m \Rightarrow (f_i, e_k), i, j, k, m \in \{1,...,n\}$.
   This rule will be denoting by *functional-transitivity rule*.

■

In this way a *functional transitivity* can be establish between a function f_i and extended set of tuples of related data.

In order to extend our definitions to allow the 'construction' of factors of processes or data it is possible to define other relations based on the above ones. The factors will allow the definition of 'equivalence classes' of data, based on one or more functions or one or more function based on a set of data. So, we can increase the scope of the data set related to a function and reciprocal.

4.6 Definition: *Functional-Transitive* Relation *(ftr)*

Given the relations $fdrf_i$ and $auxrf_i$, defined for the function f_i, the *functiona-transitive* relation for the function f_i is defined as: $ftrf_i := fdrf_i \cup auxrf_i$.

■

4.7 Definition: Factorization of a *Functional-Transitive Relation* by a Function

Given a *functional-transitive relation* defined on a function $f_i : ftrf_i$, a set of all data concerning to f_i is defined as:

1. $ftrf_i/f_i := \{d_j \mid d_j \in \{d_1,..., d_n\} \vee d_j = e_k, j, k \in \{1,..., n\}\}$. (So far, trivial).
2. For the functions f_i and f_j is possible to define:
   $ftrf_i, f_j/\{f_i, f_j\} := \{d_k \mid (f_i, d_k) \in ftrf_i \vee (f_j, d_k) \in ftrf_j\}$.
3. $ftrf_i, f_j/f_j := \{d_k \mid (f_i, d_k) \in ftrf_i \wedge (f_j, d_k) \in ftrf_j\}$.

Now, the sets of data linked to a function or to functions are defined. In principle, this definition is trivial as far as natural derivation of the earliest definitions. ■

Important Remarks:
In addition to all sets and structures defined upon to now, we can emphasize:

1. Let the *demand-driven data* set be denoted by: $ddd := \{d_r, d_{r+1},\ldots, d_s\}$.
2. Further, let the semantic conclusion of the elements of *ddd*, be denoted as: $\{d_j ,\ldots, d_h \} \models d_i$ for each $d_i \in ddd$ $(i \in \{r, r+1,\ldots, s\})$.
3. Based on the AS-IS organizational-processes model the ordered set of the functions can define a process: $P_1 := \{f_1, f_2,\ldots, f_n\}$. (If P_1 has parallel sub-processes then, we will have, for example, $P_1 := \{\{f_1,\ldots, f_k\}, \{f_2,\ldots, f_m\},\ldots\}$. But this is not the aim of this paper, therefore, we assume the processes as linear sequences of functions.).

4.8 Definition: Factorization of a *Functional-Transitive Relation* by a Set of Data.

Given a functional-transitive relation ftr_F defined on a set of functions F := { f_1 , f_2 , , f_k }, the set ddd := {d_r, d_{r+1},…, d_s} . Then, for each $d_i \in ddd$ and $\{d_j ,\ldots, d_h \}$ d_i is valid, then $ftrF/\{d_j , d_n\} := \{ f_j \mid f_j \in F \wedge (f_j , d_m) \in ftrF , m \in \models \{j,\ldots,n\}\}$ is the set of the functions which provide data for d_i . ■

In the sense of the IPD, the sets defined by Definition 0 will be the bases for the TO-BE models (the new EPC and respective data) integrated with the *Demand-Driven* data set.

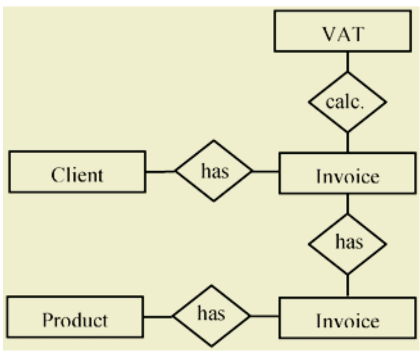

Fig. 3. Invoice system example

5 Example

In this example we will describe an invoice system. This system has an initial ERM (data-driven approach), see figure 3.

This ERM has 5 entities, containing the following data:

- Client entity – d_1 : client code; d_2 : client name; d_3 : client address; d_4 : city; d_5 : phone; d_6 : fax; d_7 : tax number; d_8 : total invoice in period.
- Invoice entity - d_{10} : invoice number; d_1 : client code; d_2 : client name; d_3 : client address; d_4 : city; d_7 : tax number; d_{30} : invoice total.

- Invoice-line entity - d_{10} : invoice number; d_{20} : product code; d_{21} :product net value; d_{22} : product VAT code; d_{23} : calculated VAT value (based in d_{21}); d_{24} : product total value.
- Product entity - d_{20} : product code; d_{41} : product description; d_{21} : product net value; d_{43} : stock quantity; d_{23} : calculated VAT value (based in d_{21}).
- VAT - d_{20} : product code; d_{21} : product net value; d_{22} : product VAT code; d_{23} : product VAT code (based in d_{21}); d_{24} : product total value.

This system has 5 functions: f_1 verification of client data; f_2 create invoice head; f_3 VAT (Value Add Tax) calculation; f_4 verification of product data to invoice-line; and f_5 invoice print.

Assume f_1 as function to verify and load the client data. This function manipulates the following data:

1. d_1 : client code.
2. d_2 : client name.
3. d_3 : client address.
4. d_4 : city.
5. d_5 : phone.
6. d_6 :fax.
7. d_7 : tax number.
8. d_8 : total invoice in period.

Assume f_2 as function to create and process an invoice head. This function manipulates the following data:

9. d_{10} : invoice number
10. d_1 : client code.
11. d_2 : client name.
12. d_3 : client address.
13. d_4 : city.
14. d_7 : tax number.

Assume f_3 as function to calculate VAT. This function manipulates the following data:

15. d_{20} : product code.
16. d_{21} : product net value.
17. d_{22} : product VAT code.
18. d_{23} : calculated VAT value. (based in d_{21}).
19. d_{24} : product total value.

Assume f_4 as function to verify and load the product data to the invoice-line. This function manipulates the following data:

20. d_{10} : invoice number
21. d_{20} : product code.
22. d_{41} : product description.
23. d_{21} : product net value.
24. d_{43} : stock quantity.
25. d_{22} : product VAT code.
26. d_{21} : product net value.
27. d_{22} : product VAT code.
28. d_{23} : calculated VAT value (based in d_{21}).
29. d_{24} : product total value.
30. d_{30} : invoice total.

Assume f_5 as function to print the invoice. This function manipulates the following data:

31. d_{10} : invoice number
32. d_1 : client code.
33. d_2 : client name.
34. d_3 : client address.
35. d_4 : city.
36. d_7 : tax number.
37. d_{20} : product code.
38. d_{41} : product description.
39. d_{21} : product net value.
40. d_{22} : product VAT code.
41. d_{23} : calculated VAT value (based in d_{21}).
42. d_{24} : product total value.
43. d_{30} : invoice total.

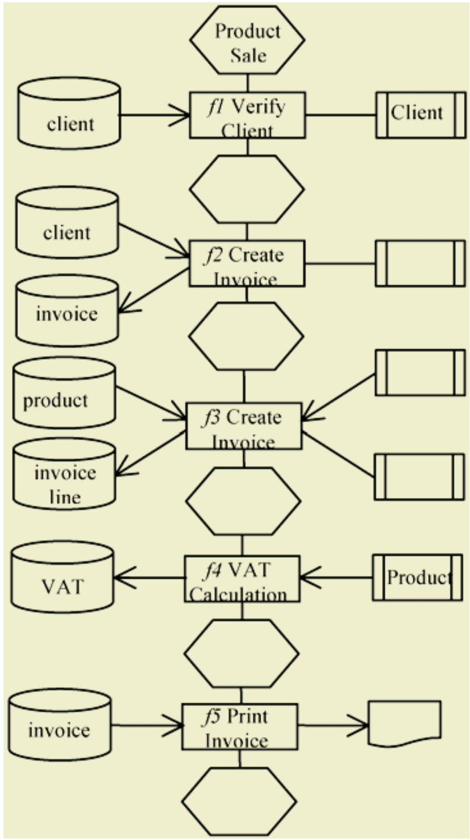

Fig. 4. description of p_1

Based on these 5 functions, we can describe the process p_1 (sale a product) which is a sequence of f_1, f_2, f_3, f_4, and f_5 , see figure 4.

The aim of the next step is gather user requirements (demand-driven approach). As result of this step we will obtain two user-requirements: compare sales information by region; and the accumulation of invoices by client and product, see figure 5. Region can be obtained through the zip code data, which is included in the data d_3 : client address. So we include two functions *f6* and *f7*, the order of this functions .

Now we can achieve a final ERM changed by IPD, see figure 6.

As demonstrated, the differences between the initial and final ERM (see figure 3 and 6) are obtained from a process p_1 with a sequence of functions [f_1,f_2,f_3,f_4,f_5] and two additional functions added by IPD *f6* and *f7*. These differences are justified through a very well defined sequence of processes – the EPC. Thus, it was started for describing the initial ERM (data-driven approach) where we obtain the AS-IS model. Its integration with the processes was demonstrated through the model EPC of the ARIS. Based on the requirements of the DW end-users (demand-driven approach) we have the TO-BE model, shaped, one more time, through model EPC of the ARIS. The differences between the TO-BE and AS-IS models, would have to be verified by the

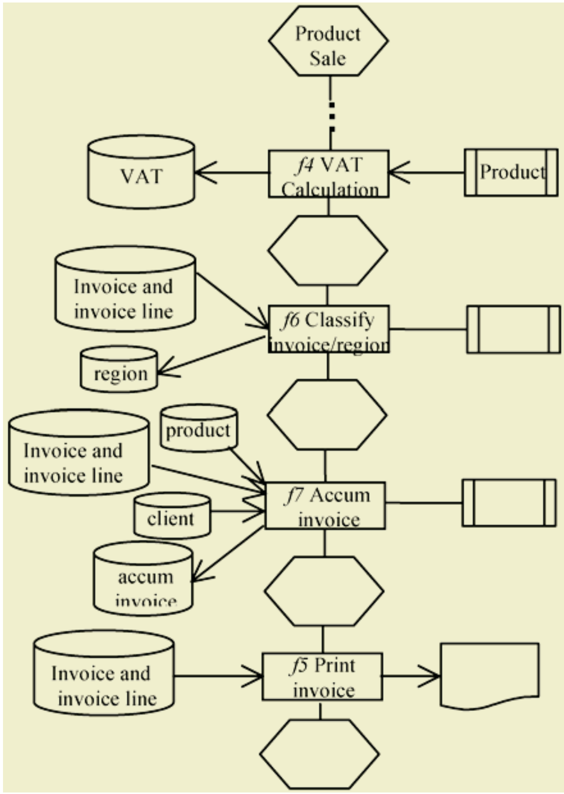

Fig. 5. EPC modified by IPD

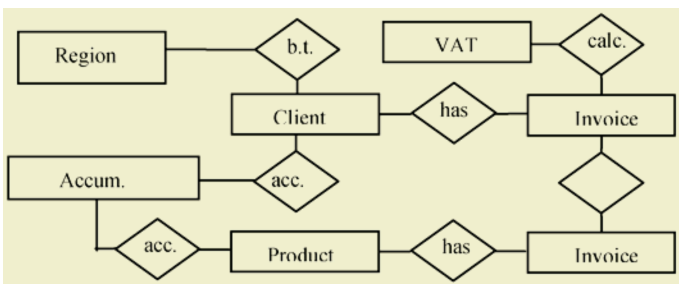

Fig. 6. ERM modified by IPD

existing goals of business (goal-driven approach) but, for the dimension of the example, it is not justified.

We have a new model ERM (figure 6). This model facilitates the design of a DW system (and respective operations to load data, usually named ETL - Extraction, Transforming and Loading) once this model include more data than the original model and, by regarding our example, more data is available to supply the needs expressed by Demand-Driven end-users. It’s important to understand that IPD, described above, can be repeated for diverse times - iterations, so the model ERM final could be the ERM initial for a similar process.

6 Conclusion

With the presented proposal we can include the organizational-processes in a DW system methodology. Since organizational-processes generate data to the DW system, these organizational processes will have to suffer a "re-engineering process", in order to satisfy the demand-driven approach. The data-driven approach only supplies part of these information's; the missing part of information would not have any relation with the organizational processes. Our proposal aims to put an end to this lack of relation between the new information and the organizational processes to get a new model of data (ERM), as well as new models of organizational processes [15].

With this approach the fundamentals of the DW methodologies have a more integrated component with the organizational processes. The IPD will enrich the DW theory with a more rigorously requirements gathering to design the DW.

In terms of research based on IPD approach, we intend to get the data model (ERM) of the DW system. By further research, we will want to framework this approach into new definitions of information systems and integrated information systems, as well as the definition of relations of congruence for the IPD to define an order, sequence of data transformations in organizational processes, with the aim to define a high degree of information integration.

References

1. Boehnlein, M., Ulbrich vom Ende, A.: *Business Process Oriented Development of Data Warehouse Structures*, In: Proceedings of Data Warehousing, Physica Verlag (2000)
2. Connelly, R. A., R. McNeill and R.P. Mosimann: *The Multidimensional Manager*, Ottawa: Cognos Inc (1999)
3. Elmasri, R.; Navathe, S. B.: *Fundamentals of Database Systems*, 3ª ed., Addison-Wesley, Massachusetts, EUA (2000)
4. Gardner, S.: *Building the Data Warehouse*, Communications of the ACM, vol. 41, no. 9, 52-60 (1998)
5. Gable, G.; Stewart, G.: *SAP R/3 Implementation Issues for Small to Medium Enterprises*, Information Systems Management Research Centre, Queensland University of Technology, Brisbane, Australia. (g.gable@qut.edu.au, g.stewart@qut.edu.au) (2000)
6. Gable, G.; Scott, J. E.; Davenport, T. D.: *Cooperative ERP Life-Cycle Knowledge Management,* Information Systems Management Research Centre, Queensland University of Technology, Brisbane, Australia. (g.gable@qut.edu.au) (2000)
7. Golfarelli, M.; Maio, D.; Rizzi, S.: *Conceptual Design of Data Warehouse from E/R Schemas*, Proceedings of the 3th Hawaii International Conference on System Sciences, Kona, Hawaii; EUA (1998)
8. Golfarelli, M.; Maio, D.; Rizzi, S.: *A methodological Approach for Data Warehouse Design*, Proceedings of the 1st International Workshop on Data Warehouse and OLAP (DOLAP'98), Washington DC; EUA (1998)

9. Immon, W. H.: *Building the Data Warehouse*; 2nd Ed., Wiley Computer Publishing, EUA (1996)
10. List, B.; Schiefer, J.; Tjoa A M.; Quirchmayr, G.: *Multidimensional Business Process Analysis with the Process Warehouse*, In: W. Abramowicz and J. Zurada (eds.): Knowledge Discovery for Business Information Systems, Kluwer Academic Publishers (2000)
11. Kaldeich, C.: *An algebraic approach to the Information Systems Integrated Theory: The binary relation (function, data).* 13ª Jornadas Hispano-Lusas de Géstion Científica, Universidade de Santiago de Compostela, Lugo, Espanha, Fevereiro (in portuguese) (2003)
12. Kaldeich, C.: *A Mathematical Method for Refinement and Factorisation of Relational Databases*; International Conference on Information System Concepts - ISCO 3 (IFIP), Marburg, R.F.A. (1993)
13. Kaldeich, C.: *Congruence relations in relational databases: incomplete information*; The 2nd Workshop on Non-Standard Logic and Logical Aspects of Computer Science - NSL'95, Irkutsk, Russia (1995)
14. Kaldeich, C.: *Toleranz- und Kongruenzrelationen in Relationalen Datenbanken*; Ed. INFIX, Sankt Augustin, R.F.A (1996)
15. Kaldeich, C.; Sá, J.: *Data Warehouse to Support Assembled Cost Centres (diagonals)*, 7º Congresso Brasileiro de Custos, Universidade de Pernambuco, Recife, Brasil. (in Portuguese) (2000)
16. Kimball, R.: *The Data Warehouse Toolkit: Practical Techniques For Building Dimensional Data Warehouse*, John Wiley & Sons (1996)
17. Kirchmer, M.: *Business Process Oriented Implementation of Standard Software*: *How to Acthieve Competitive Advantage Quickly and Efficiently*, Berlin, Springer (1998)
18. Poe, V.: Building a Data Warehouse for Decision Support, Prentice Hall (1996)
19. Scheer, A.-W.: *Business Process Engineering. Reference Models for Industrial Enterprises*, 2ª ed., Springer-Verlag, Berlin (1994)
20. Scheer, A.-W.; Nüttgens, M.: *ARIS Architecture and Reference Models for Business Process Management*, in: van der Aalst, W.M.P.; Desel, J.; Oberweis, A.: Business Process Management - Models, Techniques, and Empirical Studies, LNCS 1806, Berlin et al., pp. 366-379 (2000)
21. Watson, H.; Haley, B.: *Managerial Considerations*, In Communications of the ACM, Vol.41, No. 9 (1998)
22. Westerman, P.: Data Warehousing using the Wal-Mart Model, Morgan Kaufmann (2001)