Data Warehouse Methodology: A Process Driven Approach

Claus Kaldeich
Auxiliary Professor
cka@dsi.uminho.pt

Jorge Oliveira e Sá
Assistant
jos@dsi.uminho.pt

Universidade do Minho
Escola de Engenharia
Departamento de Sistemas de Informação (DSI)
Campus de Azurém
4800-058 Guimarães
Portugal

Fax.: +351 253 510 300

Main Keywords

Data Warehouse; Business-Processes Modelling; Integrated Informations Systems (IIS).

Data Warehouse Methodology: A Process Driven Approach

Abstract

The current methods of the development and implementation of a Data Warehouse (DW) don't consider the integration with the business-processes (organizational-processes and theirs 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 new approach will be based on the integration of organizational processes and theirs data, denote by: Integrated-Process-Driven (IPD). The principles of this approach are based on the relationships between business-processes and Entity-Relationship-Models (ERM) (the data models of the Relational Database (RDB)). These relationships come from the methodology Architecture of Integrated Information Systems (ARIS). IPD will use the information comes from the data-driven, on the one side, to match (or define) the AS-IS business processes model. On the other side, IPD will use the information comes 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, comes from the TO-BE business processes model, with the DW requirements. The aim of IPD, is to define (or 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 achieve the business goals.

Keywords

Data Warehouse, Entity-Relationship Model (ERM), demand-driven, data-driven, goal-driven, Event-Process Chain (EPC), Business-Processes Modelling, Integrated Informations Systems (IIS), Requirement-Engineering, Universal Algebra, Relational Models.

1. INTRODUCTION

Data warehouse (DW) systems have become an essential component of decision support systems in organisations. Data warehouse systems offer access to integrated and historic data from heterogeneous sources to support managers in their planning and decision-making activities. The data warehouse 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 data warehouse. The greatest potential benefits of the data warehouse occur when it is used to redesign business processes and to support strategic business objectives [WaHa98], [LSTQ00], but these are also the most difficult benefits to achieve, because of the amount of top management support, commitment, and involvement and the amount of organisational change required.

Building a DW is a very challenging issue because compared to software engineering it is quite a young discipline and 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 integration with the business processes (organizational-processes and theirs respective data). We will introduce in this paper a new approach to DW development and implementation. This new approach will be based on the

integration of organizational processes and theirs data: Integrated-Process-Driven (IPD). IPD will use the information requirements from the analysis of the operational (corporate) data model (ERM) [ElNa00] and relevant transactions – the data-driven approach, on the one side, to match (or define) the AS-IS business process model. On the other side, 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, comes 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 we discuss the three approaches to DW development methods, data-driven, goal-driven and demand-driven. In section 3, we describe the IPD approach. In section 4, we discuss the relation between processes, functions and data, based on ARIS. In section 5, we show a simple example. This paper concludes with section 6, which presents our conclusions and future research.

2. THREE APPROACHES TO DW DEVELOPMENT METHODS

Although it seems to be obvious that matching information requirements of future data warehouse users with available information supply is the central issue of data warehouse 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 approaches is to derive information requirements by analyzing business processes in increasing detail and transform relevant data structures of business processes into data structures of the data warehouse, this approach is named goal-driven. All three approaches are described in detail:

- Data-Driven (or supply-driven) approach: The data warehouse development strategy is based on the analysis of organisational data models and relevant transactions [Im96]; 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. Organisational goals and user requirements are not reflected at all [GoRi98], [GMR98]. However, these approaches risk to waste resources by handling many unneeded information structures. Moreover, it may not be possible to motivate end users sufficiently to participate because they are not used to work with large data models developed for and by specialists [Ga98].
- Demand-Driven approach: The first stage of this approach is the derivation process which determines goals and services the organisation provides to its customers. Then the business process is analysed to highlights the customers 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 [Ki96], [BoUl00]. 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.

- Goal-Driven (or user-driven) approach: This approach assumes that the organisation goal is the same for everyone and the entire organisation 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 in terms of data elements, including the definition of hierarchies [We01].

These approaches are aimed to determine information requirements of data warehouse users. End users alone are able to define the business goals of the data warehouse systems correctly so that end users should be enabled to specify information requirements by themselves. However, end users are not capable to specify their objective, unsatisfied information requirements because their view is subjective by definition, because they cannot have sufficient knowledge of all available information sources, and because they use only a business unit specific interpretation of data. Moreover, end users can often not imagine which level information the data warehouse system could supply [Ga98], [CMM99].

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 [Po96].

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 business-processes and Entity-Relationship-Models (ERM) (data models) see figure 1. These relationships come from the Architecture of Integrated Information Systems (ARIS) [Sc94], [ScNu00].

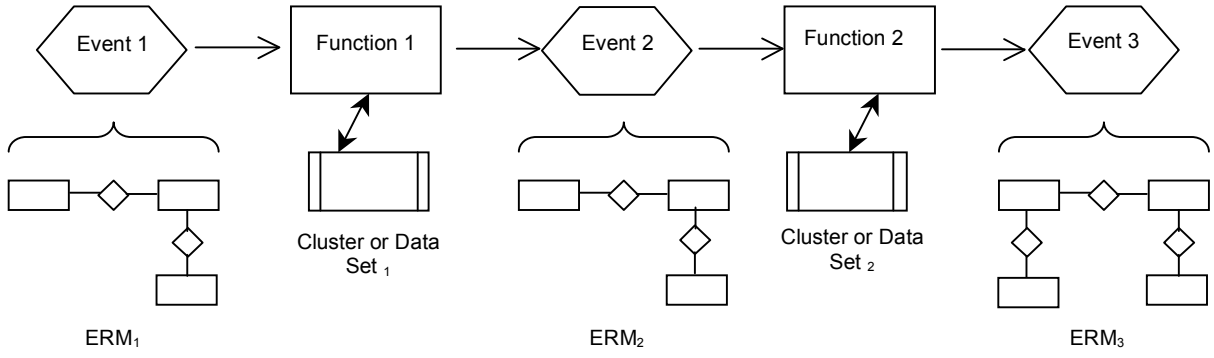

Figure 1 – Event (driven) – Process Chain

IPD will use the information comes from the data-driven, on the one side, to match (or define) the AS-IS business process model. On the other side, IPD will use the information comes 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, comes 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. 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 achieve the business goals, see figure 2.

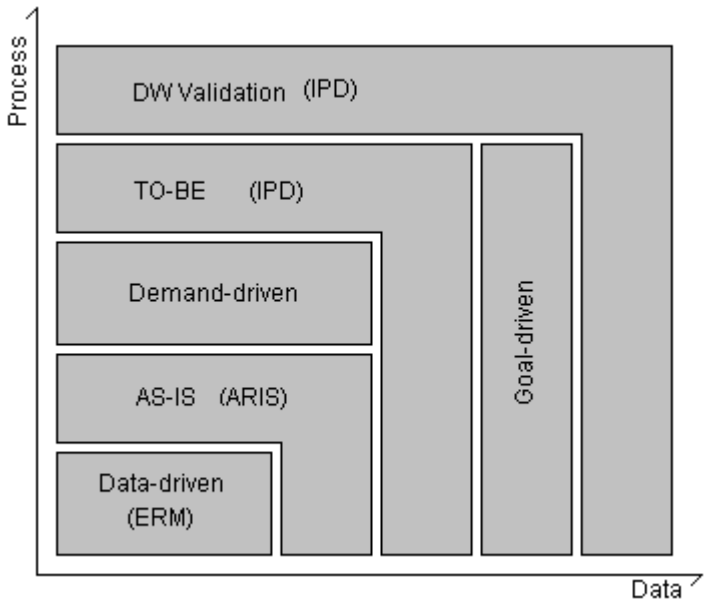

Figure 2 – IPD model

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 denote this 'combinations' or 'transformations' by *congruencies* of data [Ka95]). 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 coming from different data sets coming from different sources through complex sequences of processes.

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

Whenever we talk about *data-integration* or *process-integration* together 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 be defined the relationships between functions (as an indivisible element of a process) and respective data [Sc94].

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) [ScNu00].

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 is coming from the organizational-processes and will feed the DW and the DSS.

To support the IPD scope 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) [Ka93],[Ka95], [Ka96], [Ka03]. 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).

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*: $IC := \{\delta_i \mid \delta_i : \forall \varepsilon_j \rightarrow \psi_k;\ i,j,k \in \{1,...,n\},\ j \neq k;\ \delta,\varepsilon,\psi \in W\}$.

■

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*: $D_R := D_{at_1} \cup ... \cup D_{at_n}$.

[1] © IDS-Scheer, Saarbrücken, Germany.

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

■

Based on the definitions 4.2 and 4.3, is possible to put some questions:

a. Which following relations can be defined based on these relations to achieve the proposed goal?

b. How these above mentioned new relations complement the definitions 4.2 and 4.3?

Strictly speaking these questions derive from some simples ideas:

> *If in the definition of a relation R (definition 4.2) belongs the definition of FD_R , 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 rdf (definition 4.3) to link functions (coming 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)).*

[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.

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 4.2.
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: $fdr_{fi} := (f_i , d_j),..., (f_i , d_n))$ 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. $auxr_{fi} := \{(f_i , e_k) \mid \forall (f_i , d_j) \forall (e_m , e_k) : d_j = e_m \Rightarrow (f_i , e_k), i, j, k, m \in \{1,...,n\}$.
   This rule will be denoting by *functional-transitivity rule*.

■

In this way can been establish a *functional transitivity* 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 is possible to define others relations based on the above ones. The factors will allow to define 'equivalence classes' of data based on one or more functions or one or more function based on a set of data. So, we can enlarge the scope of the data set related to a function and reciprocal.

4.6. Definition: *Functional-transitive* Relation (*ftr*).

Given the relations fdr_{fi} and $auxr_{fi}$, defined for the function f_i , then the *functional-transitive* relation for the function f_i is defined as:

$ftr_{fi} := fdr_{fi} \cup auxr_{fi}$.

■

4.7. Definition: Factorization of a *functional-transitive relation* by a function.

Given a *functional-transitive relation* defined on a function $f_i : ftr_{f_i}$ (definition 4.6). Then o set of all data concerning to f_i is defined as:

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

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 can be defined the ordered set of the functions a process: $P_1 := \{f_1, f_2, \ldots, f_n\}$. (If P_1 have 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 $F := \{f_1, f_2, \ldots, f_k\}$, the set $ddd := \{d_r, d_{r+1}, \ldots, d_s\}$. Then, for each $d_i \in ddd$ and $\{d_j, \ldots, d_h\} \models d_i$ is valid, then $ftr_{F/\{d_j, d_n\}} := \{f_j \mid f_j \in F \wedge (f_j, d_m) \in ftr_F, m \in \{j,\ldots,n\}\}$, is the set of the functions which provide data for d_i .

■

In the sense of the IPD, the sets defined by Definition 4.8 will be the bases for the TO-BE models (the new EPC and respective data) integrated with the *demand-driven* data set.

To illustrate, from our example, the extensions of the Entity *Region*, for example the attribute *region-number*, can be derived from the extensions of the Entity *Client*, attribute *zip-code*.

5. EXAMPLE

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

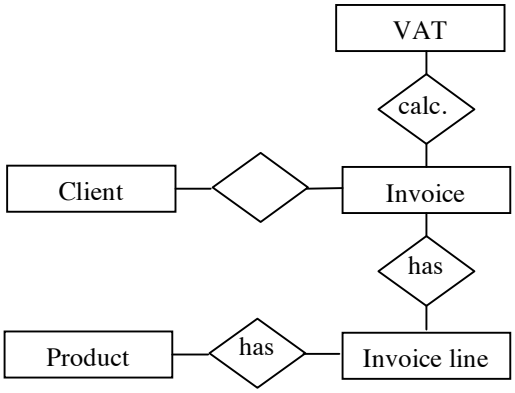

Figure 3 – Invoice system example

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 4 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; f_5 invoice print.

Let f_1 a function to verify and load the client data, this function manipulate 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.

Let f_2 a function to create and process an invoice head, this function manipulate 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.

Let f_3 a function to calculate VAT, this function manipulate 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.

Let f_4 a function to verify and load the product data to the invoice-line, this function manipulate 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.

Let f_5 a function to print the invoice, this function manipulate 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.

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.

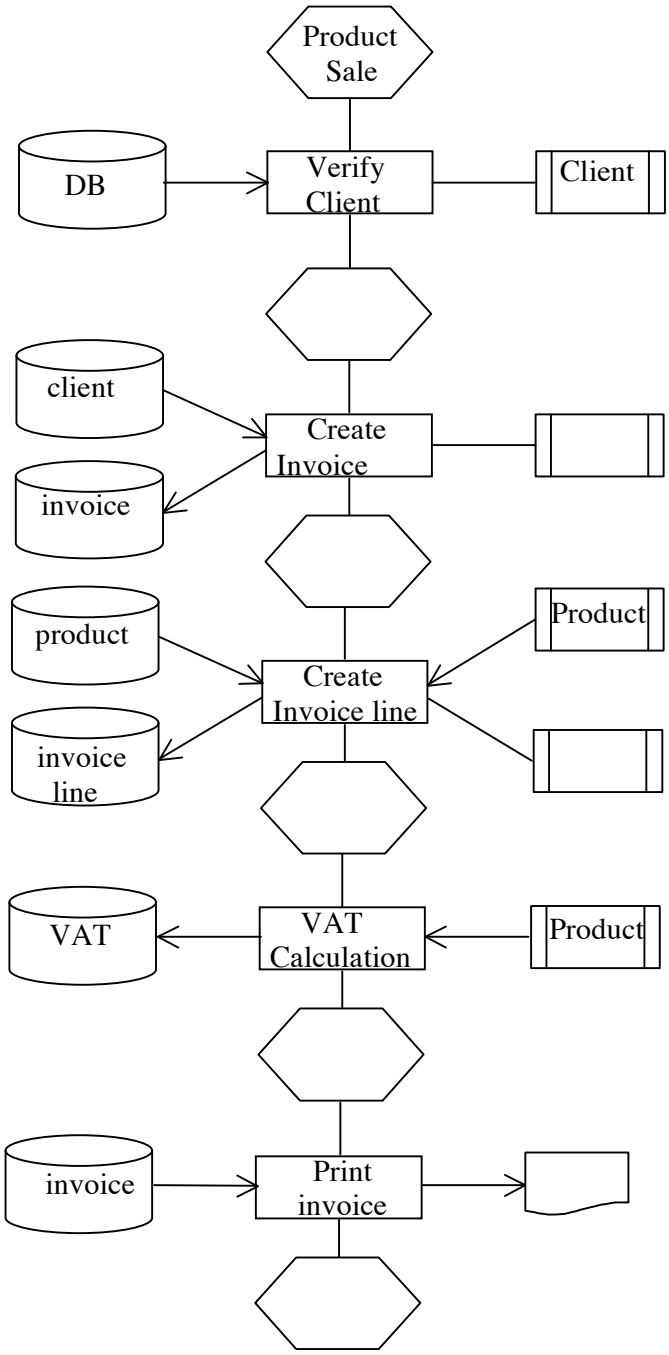

Figure 4 – description of p_1

The aim of the next step is gathering 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 are included in the data d_3 : client address.

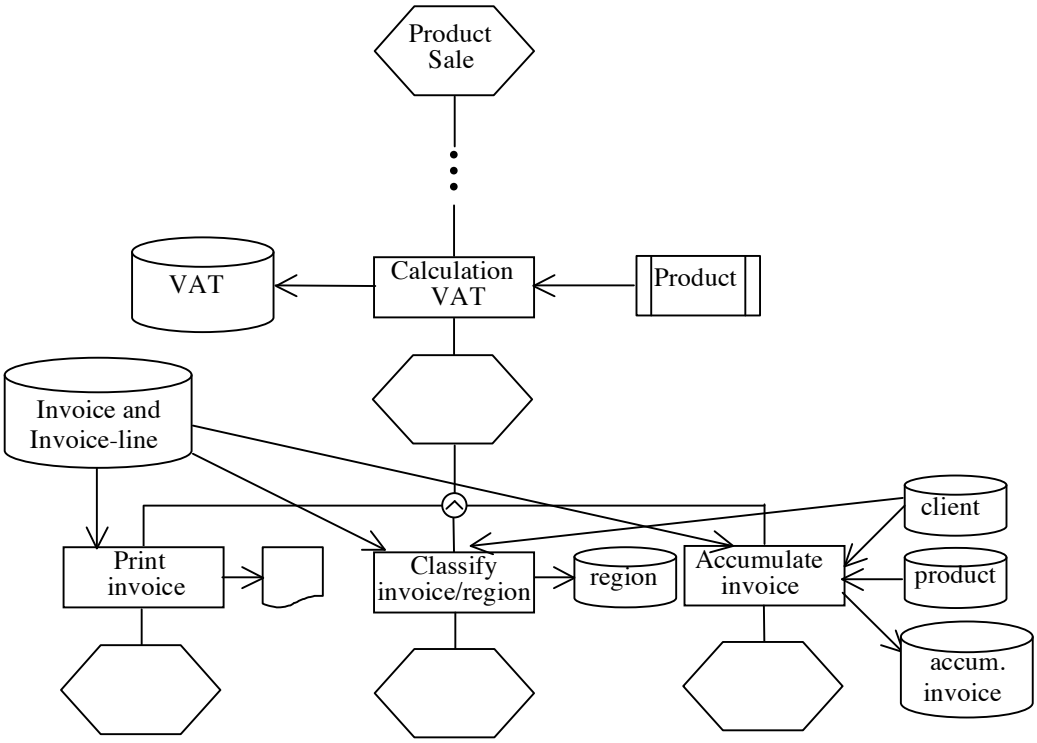

Figure 5 – EPC modified by Demand-driven

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

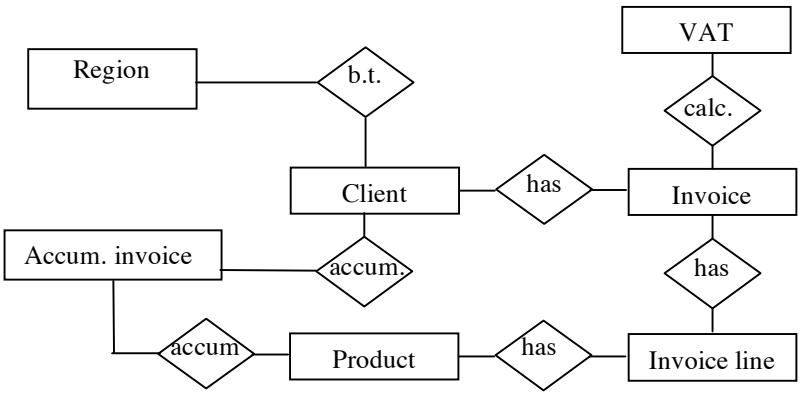

Figure 6 – ERM modified by IPD

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]. These differences are been justified by 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 got 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 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). It's important to understand that this process, 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 proposal presented 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 supply part of these informations, the missing part of information would not have any relation with the organizational processes. Our proposal has the aim to fit 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 [KaSa00].

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

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

Boehnlein, M., Ulbrich vom Ende, A. (2000) [BoUl00] — *Business Process Oriented Development of Data Warehouse Structures*, In: Proceedings of Data Warehousing 2000, Physica Verlag

Connelly, R. A., R. McNeill and R.P. Mosimann (1999) [CMM99] — *The Multidimensional Manager*, Ottawa: Cognos Inc.

Elmasri, R.; Navathe, S. B. (2000) [ElNa00] — *Fundamentals of Database Systems*, 3ª ed., Addison-Wesley, Massachusetts, EUA.

Gardner, S.(1998) [Ga98] — *Building the Data Warehouse*, Communications of the ACM, vol. 41, no. 9, 52-60.

Gable, G.; Stewart, G. (2000) [GaSt00] — *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).

Gable, G.; Scott, J. E.; Davenport, T. D. (2000) [GSD00] — *Cooperative ERP Life-Cycle Knowledge Management,* Information Systems Management Research Centre, Queensland University of Technology, Brisbane, Australia. (g.gable@qut.edu.au).

Golfarelli, M.; Maio, D.; Rizzi, S. (1988) [GMR98] — *Conceptual Design of Data Warehouse from E/R Schemas*, Proceedings of the 3th Hawaii International Conference on System Sciences, Kona, Hawaii; EUA.

Golfarelli, M.; Maio, D.; Rizzi, S. (1988) [GoRi98] — *A methodological Approach for Data Warehouse Design*, Proceedings of the 1st International Workshop on Data Warehouse and OLAP (DOLAP'98), Washington DC; EUA.

Imnon, W. H. (1996) [Im96] — *Building the Data Warehouse*; 2nd Ed., Wiley Computer Publishing, EUA.

List, B.; Schiefer, J.; Tjoa A M.; Quirchmayr, G. (2000) [LSTQ00] — *Multidimensional Business Process Analysis with the Process Warehouse*, In: W. Abramowicz and J. Zurada (eds.): Knowledge Discovery for Business Information Systems, Kluwer Academic Publishers

Kaldeich, C. (2003)[Ka03] — *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).

Kaldeich, C. (1993) [Ka93] — *A Mathematical Method for Refinement and Factorisation of Relational Databases*; International Conference on Information System Concepts - ISCO 3 (IFIP), Marburg, R.F.A.

Kaldeich, C. (1995)[Ka95] — *Congruence relations in relational databases: incomplete information*; The 2nd Workshop on Non-Standard Logic and

Logical Aspects of Computer Science - NSL'95, Irkutsk, Rússia.

Kaldeich, C. (1996)[Ka96] *Toleranz- und Kongruenzrelationen in Relationalen Datenbanken*; Ed. INFIX, Sankt Augustin, R.F.A.

Kaldeich, C.; Sá, J. (2000) [KaSa00] *Data Warehouse to Support Assembled Cost Centres (diagonals)*, 7º Congresso Brasileiro de Custos, Universidade de Pernambuco, Recife, Brasil. (in Portuguese)

Kimball, R. (1996) [Ki96] *The Data Warehouse Toolkit: Practical Techniques For Building Dimensional Data Warehouse*, John Wiley & Sons

Kirchmer, M. (1998) [Kirc98] *Business Process Oriented Implementation of Standard Software*: *How to Acthieve Competitive Advantage Quickly and Efficiently*, Berlin, Springer.

Poe, V. (1996) [Po96] Building a Data Warehouse for Decision Support, Prentice Hall

Scheer, A.-W. (1994) [Sc94] *Business Process Engineering. Reference Models for Industrial Enterprises*, 2ª ed., Springer-Verlag, Berlin.

Scheer, A.-W.; Nüttgens, M. (2000) [ScNu00] *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

Watson, H.; Haley, B. (1998) [WaHa98]: *Managerial Considerations*, In Communications of the ACM, Vol.41, No. 9

Westerman, P. (2001) [We01] *Data Warehousing using the Wal-Mart Model*, Morgan Kaufmann