Information Systems: From the Requirements to the Integrated Solution

José Francisco Zelasco and Judith Donayo
Facultad de Ingeniería, Universidad de Buenos Aires
Argentina

1. Introduction

Database integrity of an information system from its beginning to the end of its life cycle is an important issue of concern among specialists (Melton & Simon, 2002) (Guoqi Feng et al, 2009), (Post & Gagan, 2001) (Eastman C. et al, 1997) (An Lu & Wilfred Ng, 2009). The proposal solution, concerning all the aspects of an information system project, is introduced here with the aim of ensuring the integrity of the database throughout the development of the system. The general aim of this chapter is to propose a method derived from MERISE (Tardieu et al, 1985), consisting of an interconnected set of tools and those heuristics to improve the requirements engineering issues, facilitating the design of specifications, allowing alternatives of organization in terms of workstation, tasks, etc. Establish the requirements for the development of a computer system involves collecting information and expectations from users of various levels of responsibility and belonging to areas that may have, if not conflicting, at least different interests. However, the demands of different users, should reconciled into a set of specifications that will be acceptable, in a concerted way, for all of them. It is essential, to fix up the final solution, to present the proposed options in an understandable way to all users (Zelasco & Donayo, 2011).

Consequently, the information produced must be stricter and simpler, and should facilitate, in terms of design, the use of tools such as those proposed by the Unified Modeling Language (UML) and the Unified Process (UP) (Zelasco et al, 2007). In this presentation we will lay special emphasis on those tools that are related to data structuring and that make their monitoring easier during the optimization and the distribution of data on the physical level, while protecting their consistency.

As an introduction to the process of conception, we will introduce a diagram called sun (Figure 1) (Tardieu et al, 1985) in which we can see the stage of creation articulated in three levels of abstraction:

1. Conceptual level: what the company does, as a whole, to answer the external actor stimuli.
2. Organizational or logical level: (namely, involving internal actors), who does what, where (workstation) and when.
3. Operational or Physical level: how it is done and with what equipment. There is a distinction here between the tasks performed by men known as men's tasks, which give

rise to the user’s manual and the tasks performed by machines known as machine tasks, leading to the information system (programs) Design and Development.

These diagrams goes from bottom left to bottom right like the movement of the sun, passing in the middle through the upper level, i.e., the conceptual one. The entire line can be covered by iterating twice.

The first iteration occurs after the selection of elements that due to their volume (data) and frequency (events) are of greater importance. This selection is, thus, known as a representative subset and corresponds to a preliminary study of the project. The second one comprises the field of study as a whole, so it corresponds to a complete and detailed study of the system.

The line (fig. 1) from the beginning to the current conceptual level corresponds to the inverse engineering, which involves the scenarios of the system in its current state: it is the way the system is working.

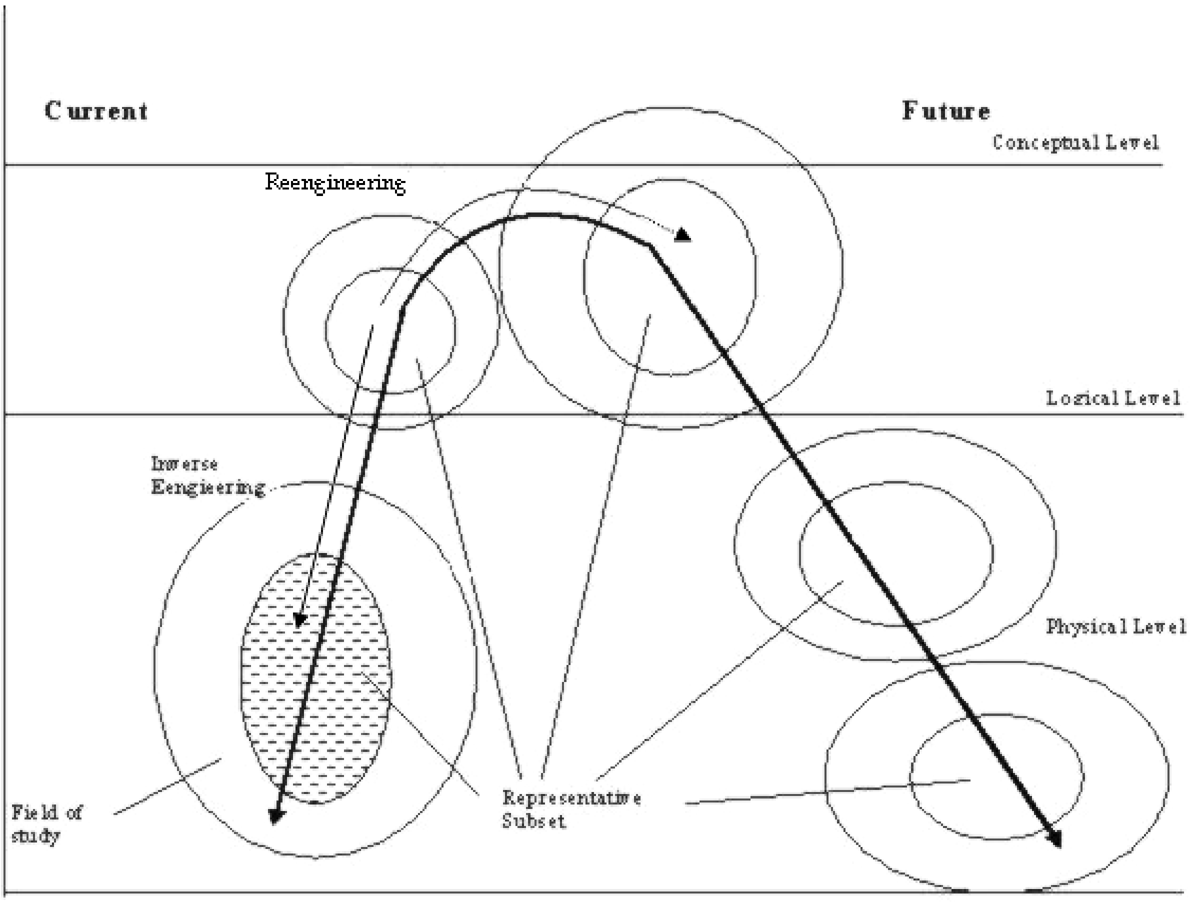

Fig. 1. Diagram of the sun

The reengineering phase involves the transition from the current state to the future state. Some of the factors taken into account for this passage are: the aim of the company, the external actors that affect it (competition, legislation, etc.) factors that could eventually have incidence in the conduct of the organization, all the Board decisions (the company's corporate rules, policies, strategies, objectives, and available resources), fields of activity, the different duties that arise from the organization chart, etc. The direction will fix up in one hand the updated management rules, involving what the company should do as a response to external requirements (stimuli); and in the other hand, the updated data integrity constraints and restrictions, will determine the data structure and functional dependencies. In the lexicon used by this method, the data integrity constraints and restrictions concern the data structure; management rules concern to the data-treatment: process and procedures -some authors use the terms Business Rules for integrity constraints and/or Management Rules together; to avoid confusion, we will not use it-. This updated information determines the passage from present state to future state, which corrects and enriches the results of the current conceptual model. The results of the current model, obtained by means of the inverse engineering, are the starting point, followed by the gathering of information about the context and definitions of the Board (reengineering) to obtain the future model.

It supported the hypothesis that there is greater invariance in the data, related to properties and classes or entities, integrity constrains and restrictions than in the treatments, concerning events and management rules. From now on, we will try to describe how to determine:

1. 1. The minimal data model that includes all the information of a distributed system, in terms of properties, entities, relations, integrity constrains and restrictions, and which will be stored in the physical database to be reached through different transitions, mechanisms and optimizations.
2. 2. Integrity verification. From the system analysis we pass on to the design, and once the relevant objects are created, the consistency between the persistent properties and the minimal established scheme is verified. This is to be done by updating and querying each one of the corresponding entities/properties. This heuristic ensures that the minimal data scheme meets the needs of each subsystem, but the main advantage of this mechanism is to ensure that each subsystem provides all the elements required by the other subsystems. In this way the modifications of the previous subsystems are to be minimized when the following ones are developed according to the priorities established by the Board.
3. 3. The treatments model based on Petri nets (Wu et al, 2005) (Chu et al, 1993) are executed by subsystem. A first scheme describes the Process itself, i.e., what the company is to do as a response to each external requirement, to initial events of processes and to those which derive from them. This brings about a concatenated series of operations, an ongoing activity, per Process. The operations contain tasks expressed in terms of management rules. Next, that same scheme is enlarged in the Procedure, replacing each operation in one or many phases, corresponding to the uninterrupted activity of a specific workstation. This brings about different organization options i.e., scenario options that, in this level, could be evaluated comparatively by the company's Board and other users, and so, to choose the most convenient one. The scheme describes the future scenarios and use cases. The following level is the operational level, in which each job position distributes the tasks according to the human's activity and the information systems activity. The tasks performed by a person will lead to the user's manual and those automated correspond to the system analysis. The tasks derived from management rules are expressed less colloquially and eventually more formally.

2. Conceptual data model

The minimal and complete conceptual data model allows an overall view of the information system data. This overall view of the memory shared by all the actors of the system (all the subsystems) is the mechanism that allows a truly systemic approach of the project. The database administrator transfers this minimal and complete conceptual model without redundancy, to its physical form through passages and optimizations.

To create the minimal conceptual data modeling, that allows for a global view of the information system, we do not proceed as in object creation, i.e., the starting point are the fields of activity from which the relevant properties are gathered and classified, trying not to confuse or relate classes to objects. This yields greater objectivity to the integrity verification, as the person who executes the data scheme is not usually the one who creates the objects of each subsystem.

During the preliminary study, (first iteration) the most important data are chosen, taking into account the volume and the most frequent processes. It is important to determine a representative subset which will allow a good articulation between the Conceptual Data Model, and all of the subsystems. This can be done iteratively, since at the stage of detailed study (second iteration including the complete set of data) such model will have all the properties with no redundancy that have to be stored in the physical base.

A list of integrity constraints and restrictions should be created to give rise to functional dependencies, relations between entities, etc. From the current model and taking into consideration the modifications that result from reengineering, we pass on to the future model, as we have mentioned above.

To avoid property redundancy, this scheme requires the existing relationships to be expressed without being transformed into binaries. Although the *look here* approach (MERISE proposal) (Tardieu et al, 1985) (Tardieu et al, 1987) (Mounyol) and the *look across* one (Chen, 1976) could simplify the representation of options in ternary relationships, it is useless to discuss their advantages because both of them are complementary. In fact, one chosen approach should have to be enriched in a simple way, to allows the representation of all the functional dependencies that both complementary approaches represent.

There are certain points to take into account to reach this minimal model. Some are rather elementary, others are more subtle:

1. Each property appears only once in the scheme, as a class (entity) or relation attribute.
2. The scheme should respect the second normal rule (Batini et al, 1991) (Ullman et al, 1997) (Yourdon, 1993).
3. If a property depends on the identifiers of two or more classes, it is a property of the relation that links such classes (third normal rule). (Batini et al, 1991) (Ullman et al, 1997) (Yourdon, 1993) (Ullman et al, 1988).
4. Only one value can be assigned to each property.
5. An arc that links a class with a relation cannot be optional, i.e., when there is a relation occurrence, it should be linked with an occurrence of each class. If this is not the case, it is because there are two different relations that have to be separated because conceptually it is not the same, even though in the physical level this will not be respected.

6. Only one occurrence of each class converges to each relation occurrence and reciprocally only one relation occurrence converges to each set of occurrences of each class. If this does not happen it is because the indispensable class that avoids this ambiguity has been omitted.
7. It should be verified that each class has a set of occurrences, so as not to confuse unique objects (company's Board) with data classes, which is a frequent mistake among beginners.

It should be pointed out that this minimal scheme can be the basis for a conceptual object-oriented database model scheme using simple generalization (Zelasco et al, 1998), and evaluating multiple inheritance.

3. Integrity verification or consistency

Integrity Verification or consistency allows us to ensure data integrity and the harmonic development of subsystems by reducing costs and the redundancies derived from the modifications of previous subsystems developed in order to satisfy the subsequent ones. The verification process consists of contrasting the object persistent properties of each system with the minimal and complete conceptual data model. With this minimal model we can verify all the updating and queries of each and every persistent data of each application or subsystem object; this is the "verification of the integrity" (fig. 2).

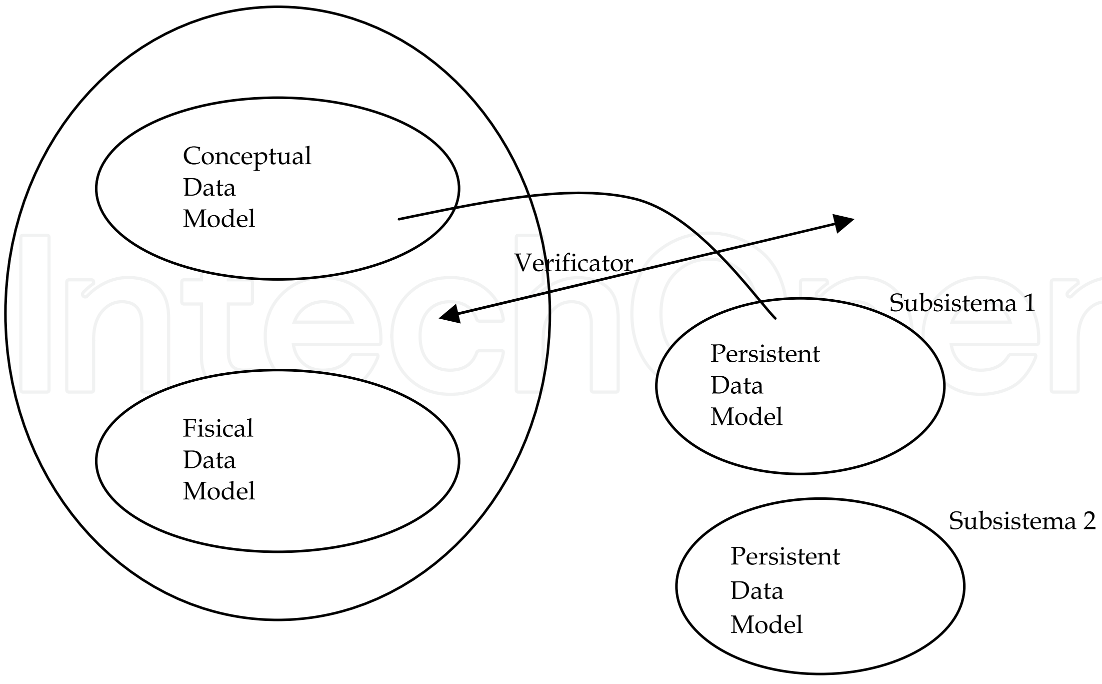

Fig. 2. Integrity verification

When analyzing the object persistent properties in previous subsystems, it is assumed that some anomalies may occur. However, in the minimal conceptual model these anomalies would be considered as a sub product of the verification process. Since verification mainly aims to ensure that each subsystem yields the essential information to the proper functioning of other subsystems. This interaction is fundamental when the information is distributed.

Some applications are developed before others for some priority reasons. In this case, the verification process guarantees that the cost of the modifications of the previously designed applications is minimum or inappreciable.

This objective is achieved by verifying that the persistent properties of each subsystem can be updated and accessed. It is neither necessary nor possible that data structure from the minimal conceptual model resembles that from the object models. However, there must be identifiers (or access paths) that allow for the updating of classes and relation occurrences as well as of particular properties.

From there on, the database administrator may choose between an object-oriented database and a relational database (Ceri et al, 1997). He will progressively simplify and include optimizations with the corresponding redundancy that will be documented to allow inverse engineering to be done without difficulty whenever necessary. This permits to control redundancy since reference is always made to the same minimal model. Indeed, whenever a modification to the database is required, (inverse engineering) that modification must be expressed in the minimum scheme so that it affects the entire distributed database to ensure data consistency. Thus, it will be enough to bring down the optimizations tree and transformations from the minimum model to the database. It should be taken into account that the optimizations must respect, if possible, the search for the minimum storage cost function and process time cost (holding cost) and it must also be well documented; thus facilitating such drop. Besides, the information will be distributed and correctly documented at the corresponding level. Relying on the documented tracing of the simplification, the optimization, and the distribution at its different levels are not irrelevant factors to guarantee integrity and the consequent minimization of the cost of development and maintenance of the project.

4. Processing model

We shall present the tools and the way to proceed with the treatments at different levels.

We suggest tools derived from Petri's nets to develop the Conceptual Model of Treatments (CMT) as well as the Organizational Model of Treatments (OrMT) (Zhang et al, 1994); (Chu et al, 1993). The richness of the proposed diagrams is superior to that of the sequence or activity diagrams and therefore more appropriate to model these levels of abstraction. As it was said, the users dispose of several options of solutions, where he can evaluate advantages and disadvantages. However, we should note that the tools supplied by UML (Larman Craig, 2001) are very useful as long as they are applied in the corresponding level.

Treatment schemes at different levels are made by considering the domains of activity the company develops and for each function which can be extracted, for example, from its organization chart. These subsystems proposed by the users and corroborated by the specialist will lead to different applications and possibly will prioritize stages of development. The activity of the company, both at the conceptual and the organizational level is reflected from the management rules, in other words, what the company should do in response to external requests (stimuli), in terms of events. Management rules of the current model are obtained as a result of reverse engineering, observing which response is given to each request of an outsider to the organization (event).

The model presented to the users is fundamental for -through reengineering, passing from the current conceptual model to the future conceptual model-, enabling the users to see in a very concise way, the changes that involve considering the new circumstances.

4.1 Conceptual model of treatments

The Conceptual Model of Treatments describes the processes that initiate from primary events which are generally external or assimilable to external ones; that is to say **what** to do.

These processes are divided into operations which have a set of management rules. The events as well as such management rules (not business rules) (Ceri et al, 1997) come from the criteria mentioned in the introduction. Management rules describe each action that the company should accomplish to satisfy events, not only the initial ones but also those arising during the process. The operations involve a series of ongoing or uninterrupted actions. The necessity of a new operation arises from an interruption, as a result of a response event from an external actor. The organization thus waits for another external event to restart the following operation. In terms of events, there is no distinction between response and events, that is to say the response is, in itself, a new event.

These events are linked to the activity sector of the organization; the events that will give rise to a particular response won't be equal in the case of a hospital or a bank or an insurance company. So it is necessary have detected all external events (or assimilable to external) that affect the organization in matter, sorted by subsystem or software application, both linked, as we said at the organizational chart. Management rules are the answers that will give the organization to each of these events. Setting these rules is a critical task because from it depend the future of the system. The management agreement is essential. It is noteworthy that different options of conceptual model of treatments, allows comparison in options of management rules and to choose one of these models implies opting for certain management rules. Management rules are codified to facilitate their use in the diagrams. The elements of process diagrams (CMT) are events, synchronization, operation or transaction with any eventual output condition and responses (see the figures in example below). The operations correspond to the performance of one or more management rules and, in a whole, will lead to response events. We insist to notice that the management rules describe each of the actions that the company must take to satisfy events, both initial and those which arise during the process that will involve an operation or a chain of operations. The operations engage a series of uninterrupted actions. The need of a new operation occurs due to an interruption, as a result of an event-response to an external actor, remaining waiting for another external event to reset the next operation. No distinction is made in terms of events between response and event, i.e. the answer is, at the same time, a new event.

It is noteworthy that at this level (CMT), it ignores who does what, when and where in the organization of the company. The responses to events are made by the company as a whole. This allows in the subsequent level, propose, analyze and evaluate different options of organization having previously set the response of the company as a whole in front of each of the external events. In this CMT is observed the synchronization, prior to each transaction, the synchronization uses the logical operators of conjunction and disjunction (and and or) (Fig. 2.5). This allows representing the different sets of events (or a single event) that trigger the operation. In the case of response events, there can also be an indication of the output conditions in which they are produced. Within the operation or transaction, which must be identified with a name, it is placed the code of management rules that apply. The acronyms of the corresponding management rules are inscribed in the operation which must have a name. These acronyms are in correspondence with the list of rules where the actions involved are described.

4.2 Organizational model of treatments

The Organizational Model of Treatments (OrMT) describes the characteristics of the treatments that have not been expressed in the conceptual model of treatment, expanding, in terms of workstations, the conceptual model of the company's organization; that is to say, workstation (who do what, where and when), in other words, time, human resources, places, etc..

The conceptual model of treatments describes the flow of events, mainly those between the organization of the company and the environment (**what**).

The OrMT adds to the conceptual model the company's flow of events among the different workstations (Chu et al, 1993).

A process in the CMT expands in a procedure in the OrMT and operations of each process will result in different phases of the work stations of each procedure. The CMT describes the flow of events, essentially between the organization (the company) and the environment. The uninterrupted or ongoing activity at the procedure level is called phase. The OrMT adds for the CMT the flow of events within the company among the different phases of the work stations (see the figures in example below).

The study of a company's organizational problem, usually, belongs to other experts, so computer specialists are frequently restricted to taking that model as the working base for the system development (Dey et al, 1999). Different organization options, other than the one imposed, show improvements in the information system. This methodological proposal not only prevents this inconvenience and the waste of money and time but also proposes a study of the company's organization options setting, possibly in question, the solution proposed by other experts. When a computer specialist takes part in contributing with this formal tool, based on Petri nets (He et al, 2003), the advantages of one option over the other one become obvious, particularly from the automatization point of view. The decision about the company's organization results, then, in an agreement, and so the computer specialist faces a more solid and robust option. Besides, this OrMT contributes to a more accurate elaboration of the user's manual and to the analysis prior to design.

A chart is used to describe the procedure. The first column or the first set of consecutive columns corresponds to the external actor/actors related to such process. The subsequent columns correspond to the workstations to which the procedures are expanded. The phase structure is identical to the operation structure of the conceptual model of treatments. The phase has the possibility of synchronizing events by means of logical operators of conjunction and disjunction (and and or). It has an identification name, the acronyms of the management rules which correspond to the actions that a workstation must carry out during that phase and the output conditions of the response events. It is important to highlight that response events may be directed to both external actors and internal actors (other workstation phases). From the foregoing, it can be said that one or more phases correspond to each operation and that in the case the operation is divided into various phases, the management rules contained in the operation will be distributed among the respective phases.

Finally, in case the phase has rules capable of automatization, the operations that will be done by the person in that workstation and those operations to be automated will be established. These actions are called "men's tasks" and "machine tasks". This classification corresponds to the called Operational Treatments Model (OpTM) and gives rise to the user's manual and the systems analysis at the level of each use case.

For the OpMT it can establish an analogue diagram between man tasks and machine tasks, which can facilitate the development of the using manual and the pre-programming analysis, however, is not necessary to discuss this scheme with the user as it is an appropriate tool for the specialist.

The progressive detail grade of these three levels shows that they are three different levels of abstraction and that they are useful for a better information system definition. In this way the representation of use cases and scenarios options is facilitated.

5. Examples

This section develops a case study that focuses on the subsystem for admitting patients to an attention centre just with the purpose of presenting the development of treatment paradigms. This is not a real case, but an example that brings out the versatility of the proposed heuristics.

To be able to realize the process for the CMT identifies the events that stimulate and initiate this process and management rules, which indicate how the organization should respond to these events (Fig. 2).

5.1 Description of the admitting process

The admission process begins when a patient arrives at the center of attention. A patient that request an appointment can reach the center of attention in two conditions, as patient with health insurance in which case he pays a supplementary amount (reduced) or as a patient without insurance, in which case he pays the full benefit.

And from the solution 1 they are proposed two options of solution for the OrMT: the diagram a (Fig. 3) and diagram b (Fig. 4).

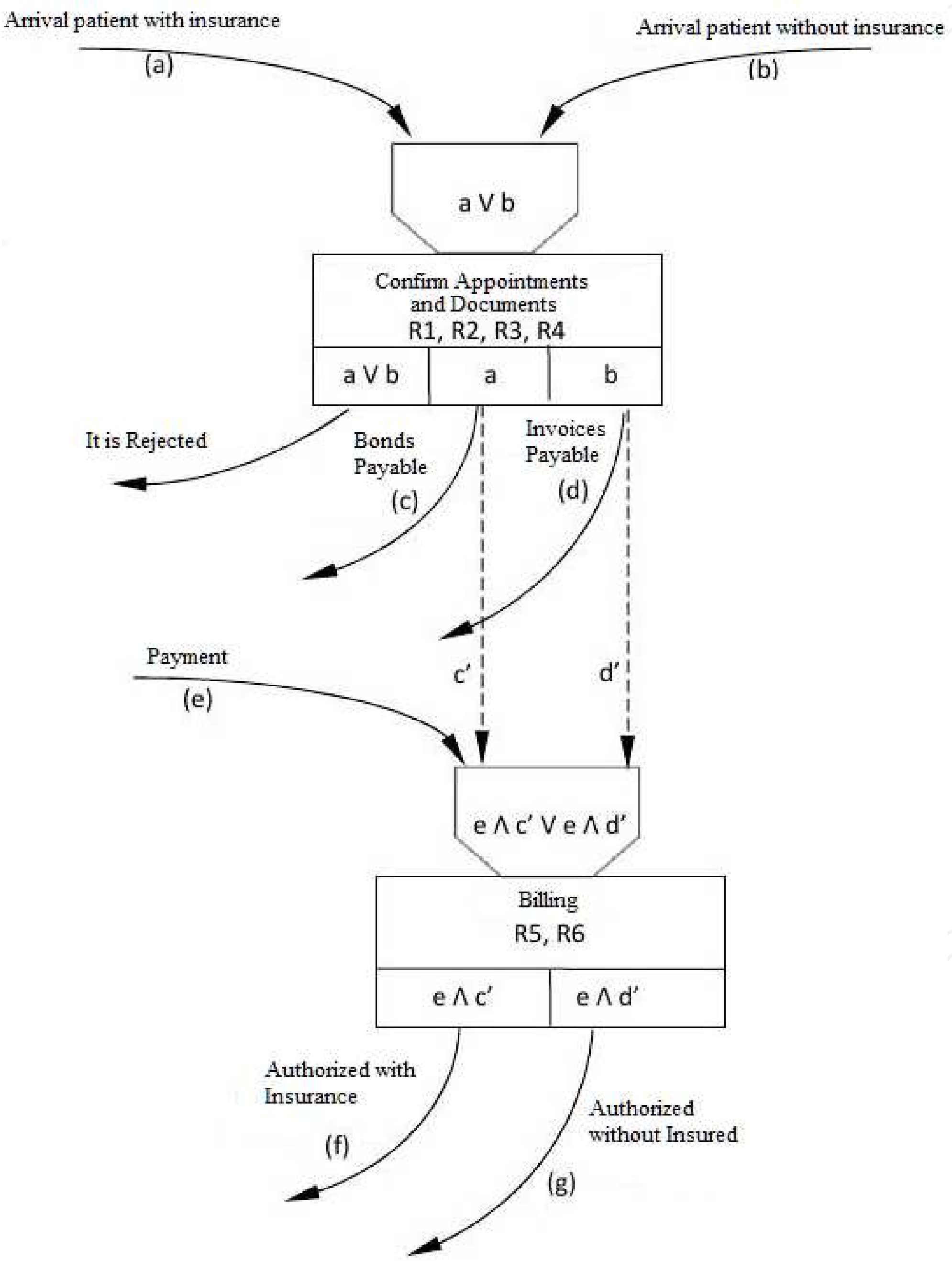
Arrival patient with insurance
(a)
Arrival patient without insurance
(b)
a V b
Confirm Appointments and Documents
R1, R2, R3, R4
a V b
a
b
It is Rejected
Bonds Payable
(c)
Invoices Payable
(d)
Payment
(e)
c'
d'
e Λ c' V e Λ d'
Billing
R5, R6
e Λ c'
e Λ d'
Authorized with Insurance
(f)
Authorized without Insured
(g)

Fig. 3. Conceptual Model of Solution 1 of the example of admission of patients.

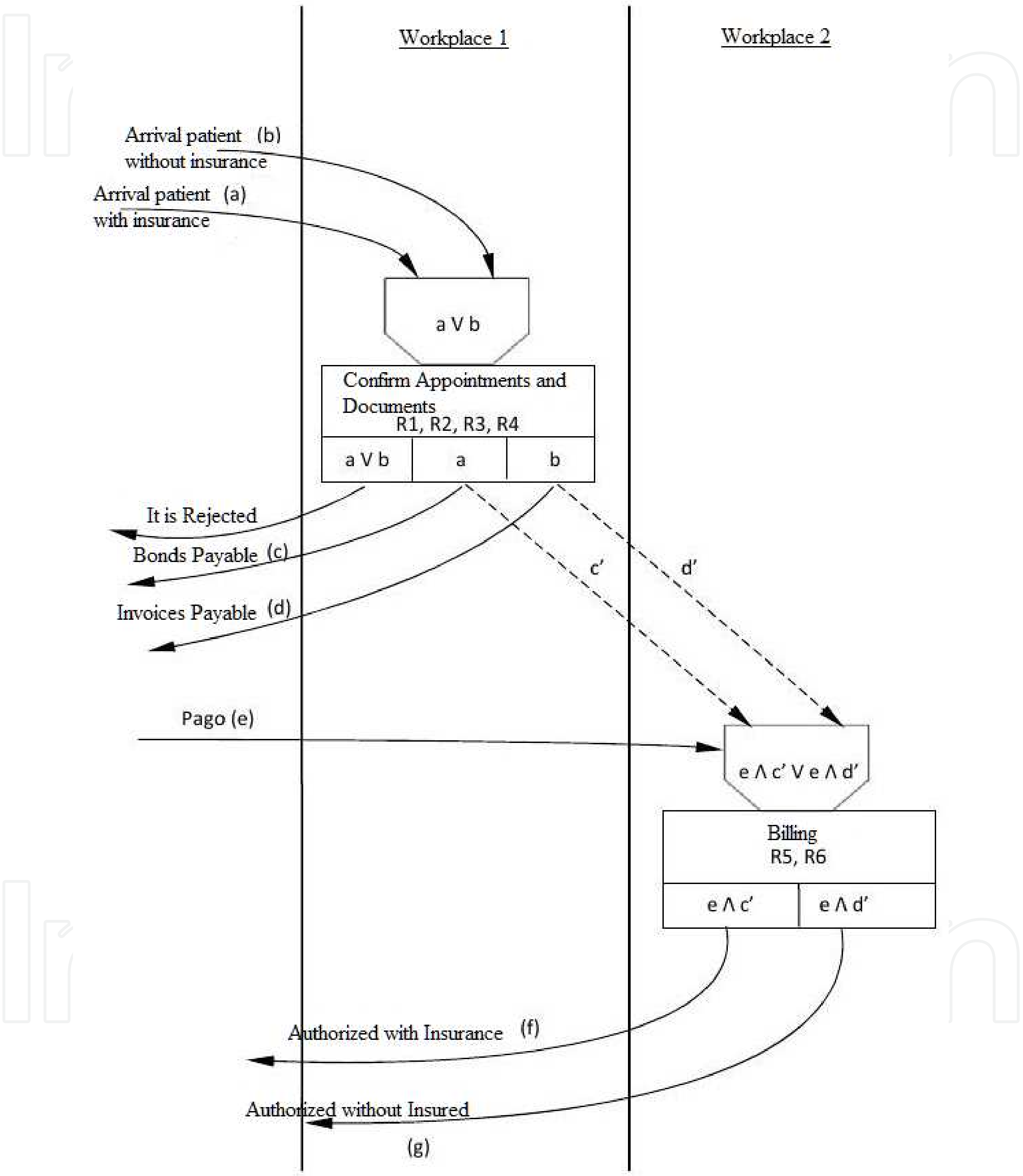

Fig. 4. Organizational Model of Solution 1 of the example of admission of patients to a health centre.

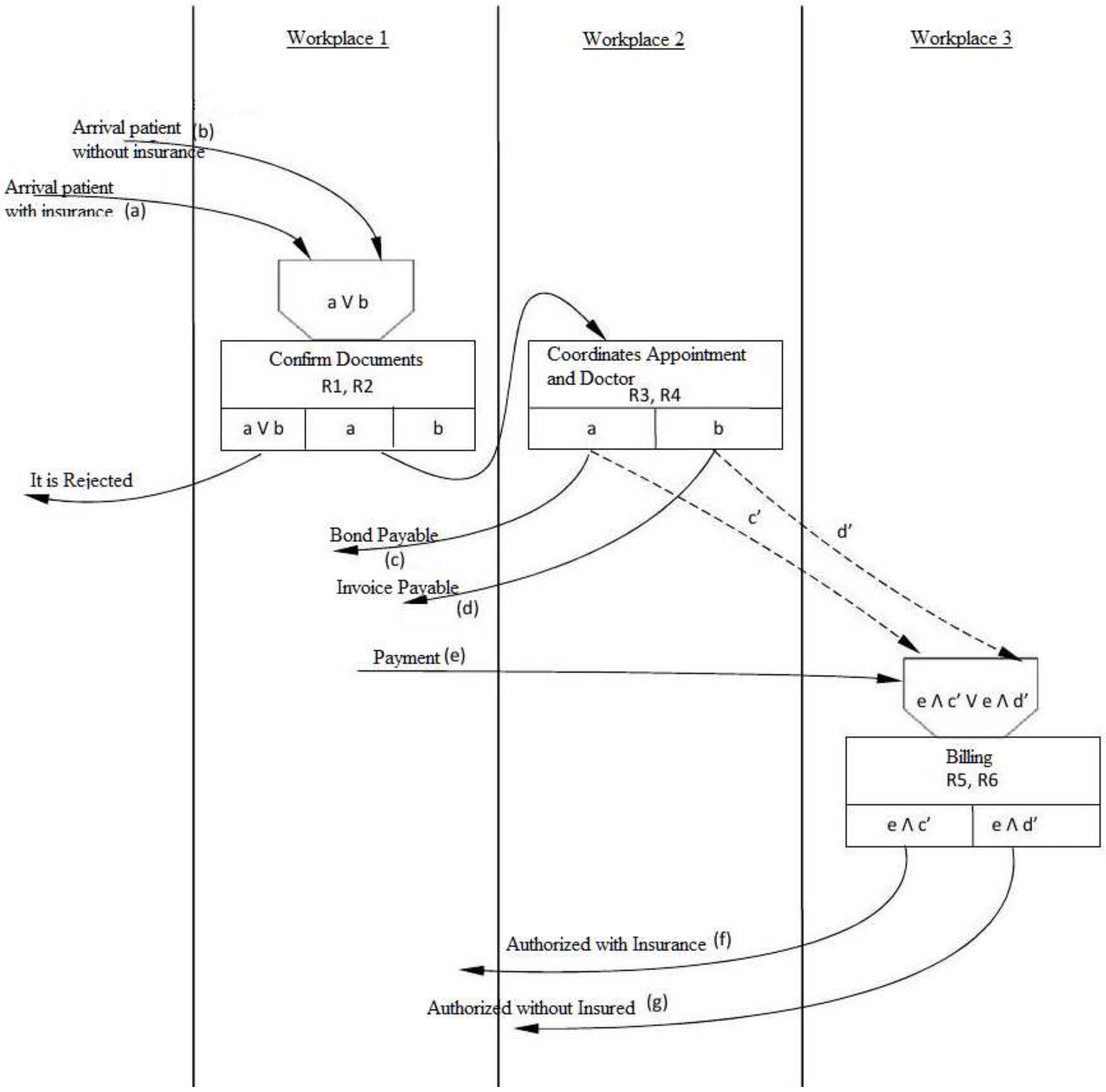

Fig. 5. Another Organizational Model of Solution 1 of the example of admission of patients to a health centre.

5.1.1 CMT Solution 1

Next, we present the external events and the management rules corresponding to the process of the solution 1:

Events:

a. Arrival a patient with insurance.
b. Arrival a patient without insurance.
c. Effectuation of payment.

Rules:

R1: Identification verification otherwise is rejected.

R2: Identify specialty and schedule an appointment with the doctor.

R3: With the appointment scheduled concerted, if the patient has insurance, the bond is issued.

R4: With the appointment scheduled concerted, if the patient hasn't insurance, the invoice is issued.

R5: Verified the bonus payment, a permit is issued to the patient with insurance, to make an appointment scheduled.

R6: Verified the invoice payment, a permit is issued to a patient without insurance, to make an appointment scheduled.

Note that the operation is interrupted because the payment can be made later, not at the same time as the application of turn.

5.1.2 CMT Solution 2

Next, we present the external events and the management rules corresponding to the process of solution 2:

Events:

a. Arrival a patient with insurance.
b. Arrival a patient without insurance.
c. Effectuation of payment.

Rules:

R1: Identification verification otherwise is rejected.

R2: Payment according to condition.

R3: Verified the payment, it proceeds to the specialty identification and it coordinates appointment scheduled, doctor and emission of authorization of corresponding consult.

It is noted that the operation is not interrupted, because the payment it has to be done at the same moment than the appointment solicitation. In this case, because the process is reduced to only one operation, the reader will be able to elaborate this process, including all the rules in the only operation.

5.1.3 Organizational models of treatments corresponding to solution1

Supposed being selected the solution 1 (Fig. 2) they are analyzed the two solution options proposed for OrMT: diagram a (Fig. 3), diagram b (Fig. 4).

In diagram a (Fig. 3), the first operation (Validation of appointments scheduled and documents) of the CMT is performed in the phase validation of appointments scheduled and documents on the first work station, and the second operation (Billing) is done in invoicing phase on the second workplace.

In the diagram b (Fig. 4) the first operation (Validation of appointments scheduled and documents) from the CMT is unfolded in the phase of "document validation" on the first work station, and the phase "coordinating of appointment and doctor" on the second work station; and the second operation (invoicing) is held in the Invoicing phase on the third workplace.

The diagram b would be selected by an organization that prefers to separate a checkpoint (security) from assigning appointments scheduled and doctors.

It is noted that the same workplace can do several phases, what happens if the procedure is more complex.

When the operational level of treatment to a certain stage of a workplace, analyzes the management rules, naturally are being established both the instruction manual corresponding to operator tasks such as system analysis derived from update tasks that should make interacting with the operator.

6. Conclusions

Preserving the minimal conceptual model along the system life preserves from inconsistencies

The minimal conceptual data model provides a systemic global view.

The current model of data and treatments, achieved by inverse engineering, guarantees a higher stability of the system.

And finally, the consistency verification assures minimal modification in previous developed subsystems.

The overall view through the minimal model permits us also to control redundancies and to avoid problems of integrity.

Applying this mechanism and consequently reducing the entropy leads to a more balanced and tolerant system to the changes in the context in which the organization is immersed and to the requirement modifications.

The application of these tools and heuristics ensures the stability of the database, provides a reduction of the number of iterations in the development thus contributing to diminish the system entropy whose cost may be inappreciable mainly at the beginning of the maintenance phase.

In the other hand, the application of this mechanism based on Petri diagrams, and respecting the indicated heuristic, is of great interest as a tool of engineering of requirements. These tools used in abstraction levels superior than usual, lead to relevant advantages. First allows better interaction between users and the specialist to formalize the requirements. The user through these schemes, of reading relatively simple, can select the most convenient solution and will compare management options when it is about CMT and organization in the OrMT. This advantage is accentuated because the proposed mechanism is performed in an iterative fashion by focusing on a representative subset in the preliminary study and taking into consideration the entire field of study in the detailed study. The development of the solution results more robust as it facilitates the visual comparison of the current model (reverse engineering), both of management and organization with different options of the future model (reengineering). The result is greater clarity in the proposed solution which gives greater rigor in the development of the specifications. In addition, from the diagrams adopted derive both the user manuals as analysis and design necessary to the development, and from comparing the diagrams of the current state with the adopted solution diagrams, arises the definition of works stations and of the intermediate phases necessary for a harmonious integration of the new system in the company.

7. References

An Lu, Wilfred Ng, 2009: Maintaining consistency of vague databases using data dependencies. Data & Knowledge Engineering, In Press, Corrected Proof, Available online 28 February 2009

Batini C., S. Ceri y S. B. Navathe, 1991, Database Design: An Entity-Relationship Approach, Prentice-Hall.

Ceri, S., Fraternali, P., 1997, Designing Database Applications with Objects and Rules: The IDEA Methodology Addison Wesley; 1st edition, ISBN: 0201403692.

Chen, P. S., 1976: The entity relationship model: to-ward a unified view of data, ACM Transactions on Data-base Systems, (1), 9-36.

Chu, f., Proth, J-M., Savi, V. M., 1993: Ordonnancement base sur les reseaux de Petri. Raport de recher-che No 1960 INRIA Francia.

Dey D., Storey V. C. and Barron T. M., 1999, Im-proving Database Design trough the Analysis of Relation-ship, ACM Transactions on Database Systems, 24 (4), 453-486.

Dullea J. and I. Y. Song, 1998, An Analysis of Struc-tural Validity of Ternary Relationships in Entity Relation-ship Modeling, Proceed. 7^{o} Int. Conf. on Information and Knowledge Management, 331-339.

Eastman C, Stott Parker D. and Tay-Sheng Jeng 1997: Managing the integrity of design data generated by multiple applications: The principle of patching. Research in Engineering Design, Volume 9, Number 3 / septiembre de 1997

FAQ Merise et modelisation de donnees - Club d'entraides developpeurs francophones. http://uml.developpez.com/faq/merise/

González, J. A., Dankel, D., 1993: The Engineering of Knowledge-Based Systems. Prentice Hall.

Guoqi Feng, Dongliang Cui, Chengen Wang, Jiapeng Yu 2009: Integrated data management in complex product collaborative design. Computers in Industry, Volume 60, Issue 1, January 2009, Pages 48-63

He, X., Chu, W., Yang, H., 2003: A New Approach to Verify Rule-Based Systems Using Petri Nets. Information and Software Technology 45(10).

Larman Craig, 2001: Applying UML and Patterns: An Introduction to Object-Oriented. Analysis and Design and the Unified Process Prentice Hall PTR; 2d edition July 13, ISBN: 0130925691.

Melton J, Simon A R, 2002 Constraints, Assertions, and Referential Integrity SQL: 1999, 2002, Pages 355-394

MERISE - Introduction à la conception de systèmes d'ınformation. http://www.commentcamarche.net/merise/concintro.php3

Mounyol, R. MERISE étendue: Cas professionnels de synthèse. ISBN: 2729895574. Rojer Mounyol ed. Elipses

Post G., Kagan A., 2001: Database Management systems: design considerations and attribute facilities. Journal of Systems and Software, Volume 56, Issue 2, 1 March 2001, Pages 183-193

Song, I. Y., Evans, M. and Park, E. K., 1995: A comparative Análisis of Entity-Relationship Diagrams, Journal of Computer and Software Engineering, 3 (4), 427-459.

Tardieu, H., Rochfeld, A., Colletti, R., 1985 La Méthode MERISE. Tome 1. ISBN: 2-7081-1106-X. Ed. Les Edition d'organization.

Tardieu, H., Rochfeld, A., Colletti, R., Panet, G., Va-hée, G., 1987 La Méthode MERISE. Tome 2. ISBN: 2-7081-0703-8. Ed. Les Edition d'organization.

Ullman J. D. and J. Windom, 1997, A FirstCourse in Database Systems, Prentice-Hall.

Ullman Jeffrey D. & Freeman W. H., 1988, Principles of Database and Knowledge-Base Systems. Vol. 1, 1a edition, ISBN: 0716781581.

Wu, Q., Zhou, C., Wu, J., Wang, C., 2005: Study on Knowledge Base Verification Based on Petri Nets. Inter-national Conference on Control and Automation (ICCA 2005) Budapest, Hungry, June 27-29.

Yourdon, Inc., 1993: Yourdon Method, Prentice-Hall.

Zelasco J. F, Alvano, C. E., Diorio, G., Berrueta, N., O'Neill, P., Gonzalez, C., 1998: Criterios Metódicos Básicos para Concepción de Bases de Datos Orientadas a Objetos. Info-Net, III Congreso Internacional y Exposición de Informática e Internet. Proceding en CD. Mendoza, Argentina.

Zelasco, J. F., Donayo, J., Merayo, G., 2007, Complementary Utilities for UML and UP in Information Systems, EATIS 2007. (ACM DL). Faro, Portugal. ISBN:978-1-59593-598-4

Zelasco, J. F., Donayo, J., 2010, Database integrity in Integrated Systems INSTICC PRESS 2009,. Milán-Italia. May 2009. ISBN 978-989-8111-90-6

Zelasco, J. F., Donayo, J., 2011, Organizational Chart for Engineering of Requirements. IASTED SE 2011 Soft-ware Engineering, Innsbruck, Austria; 15-17 February 2011

Zhang, D., Nguyen, D., 1994: PREPARE: A Tool for Knowledge Base Verification. IEEE Trans. on Knowledge and Data Engineering (6).