INFORMATION SYSTEM DESIGN WITH DYNAMICALLY CHANGING REQUIREMENTS

Belkacem Kouninef

King Faisal University

Saudi Arabia

ABSTRACT

Different semantic models that take into account the dynamic aspect of information systems, have been suggested until now. Such models make it possible to improve the conceptual schemas by representing, in addition to the structure, the behavior of the system. Among the methods that suggest a dynamic model, we can mention ACM/PCM , REMORA, DADES , IDA ,... Unfortunately, there exist only few tools able to completely support the task of design while taking into account the dynamic aspect. This paper presents a software tool which is a conceptual help to the design process, that combines the NIAM and JSD methods in order to cover not only the static aspect (data) but also the dynamic aspect (processing+behavior) of the system. The validation of the specification in a rigourous way (automatically) is done in the formalism of Rewrite Logic.

KEYWORDS: Information system design, Conceptual schema, static aspect, dynamic aspect, NIAM (Nijssen Information Analysis Methodology), JSD (Jackson System Development).

1. Introduction

The design is the essential phase in the life cycle of information system (IS). Its aim is to produce a detailed specification of information system. The design is a complicated task which could be achieved by using methodologies which are generally supported by tools. However, we can notice that the latter are themselves complicated systems. Either because they are too theoretical to be accessible to the designer, or because they are "black boxes" which produce documents and whose logic is difficult to grasp for their users (it's the case for most help design tools).

The difficulty of the design process leads us to develop a tool which gives a help in the design process, taking into account the static aspects (data) and the dynamic aspects (state change of data).

Section 2 of this paper presents design methods and their evolution. The dynamicity in the information system is introduced in section 3. Section 4 introduces existing design tools. Section 5 is concerned with the choice of the used methods. The tool is described in section 6.

2. Design methods

Since the sixties , many design methods have been developed. Their important aim in this field is to reduce the complexity of the development process. The methods initially developed are centered on the manner in which the design process is conducted.

We can quote Corig (Corig 71), SSA (Gane 89), Sadt (Ross 87), and Isac (Lundeberg 92). Since the eighties, the new methods (NIAM) (Nijssen 89), Remora (Rolland 91), IDA (Bodart 89), Merise (Tardieu 86) ...) have been based on the models. In the chronological evolution of the design methods, we can classify the methods.

We introduce two classes that we call the Cartesian method class and the systemic method class. Each class is based on a paradigm (respectively the Cartesian paradigm and the systemic paradigm) and on a specific approach of the design problems (respectively functional approach and conceptual approach). We briefly characterize the two methods.

The Cartesian methods are characterised by the way in which the design process is managed. These approaches are centered on the decomposition of the design process into phases , steps and sub-steps.

These methods propose a functional approach. In this approach, the IS is considered as an information processing system which memorises, treats, formats and communicates the data. The IS is seen as a "black box" defined by the output to produce.

For systemic methods the bearing isn't essential. In this approach the IS is viewed as a system of elements and relationships among them. The systemic methods emphasize the global aspect of IS that they analyse as a system, on the decomposition of the system into elements and on the relationships among thess elements. The methods based on a systemic approach assimilate the design process to a modelisation process; the result is an abstract representation (conceptual schema) of the real world built with the concepts of the model that was used.

Recently, the current methods suggest the integration in the conceptual schema of the dynamic aspect of the real system. These methods allow the enrichment of the conceptual schema by representing the structure and the data behaviour as ACM/PCM (Brodie 82), REMORA (Rolland 91), CIAM (Bubenko 92), TAXIS (Mylopoulos 82), IDA (Bodart 89).

3. Dynamicity in Information System (IS):

The Conception stage is the most important one in the life cycle of an (IS) . It consists in modeling the organisation reality, in other words, it needs specification about its static and dynamic aspects.

We can illustrate this category of models of the dynamicity by the three following methods: MERISE, REMORA and JSD.

(i) Dynamicity in Merise method: (Tardieu 86)

Merise has adapted the Petri Nets concepts (place, transition, token, triggering) to their use in IS (message, event, operation, result,...). The Merise method can give information about successive states of IS and sequences which allow to go from one state to another.

The used concepts are: event, result, operation and synchronisation. By analogy with Petri nets, events and results correspond to places, operations to transitions and synchronisation to trigger predicates. The event occurrences are the Petri Nets tokens.

(ii) Dynamicity in Remora method: (Rolland 91)

Remora has adopted a semantic model which allows the representation of the static and dynamic aspects at the same time in the real world.

This model has 3 kinds of fact classes: Entity, Action, Event and 4 kinds of association classes between the other 3 classes. It enables a causal description of behaviour of areal system. The model concepts are: Object, Event and Operation.

(iii) Dynamicity in JSD method: (Jean 89)

The JSD method **(Jackson System Development)** is specially designed for systems who has a behaviour changes in time. JSD specifications include sequential processes which communicate together. Sequential processes are about the behaviour of the dynamic entity.

This method is composed of 3 stages:

- modelling world
- modelling the system to be built
- implementation.

The first stage is composed of 2 steps:

- entity-action
- entity-structure.

The first step is about knowing the real world and listing the entities and the actions concerned. The second step consists of ordering the actions, which are executed or submitted, by diagrams or pseudo-codes.

The second stage modelises the system under way of a network called **SSD** (**System specification Diagram**). The processes can have relations between them by data flows or by state vectors.The data flows are like FIFO's communication and the state vectors are equivalent to local variables of processes, which can be read only **(Jean 89).**

4. The Design Tools:

At level of design phase, the existent tools are:

- Data Dictionary: (Tardieu 86)

They are especially storage and retrieval of information tools.

- Graphic Tools: Gambit (Bounadja 93)

They are interfaces which allow to grasp graphically a conception phase: GIOTTO (Tamassia 93) & RAMATIC (Dahl 95), for the case of graphic editing. GDBDA (Chan 90) allow editing grasp but without help in schema elaboration. We can cite for example:

- PC_IAST (Walfard 88) developed at Control Data which can support concept of NIAM method (Nijssen 92).

- The Software CONSOI (Dzenan 88) realised by Canadian corporation SYSTEMOID.

- The software EXCELERATOR (Perney 88) realised by American Corporation INTECH. These tools use two kinds of models: Data conceptual model (MERISE) and Data flow diagram model (Excelerator/Yourdon).

- The software MEGA of Gamma Corporation working for MEGA and MERISE methods.

5. Justification of the proposal methods:

The choice is justified by a certain duality which exists between the two specifications NIAM and JSD each in the aspect of processing. There exists a coherence between the two methods. One validates the other , so that there is a relationship between data specifications and data processing.

The NIAM method is based upon the relational binary model and disposes of an important number of constraints (exclusion, totality, equality,...) which enable to express explicitly the semantic of real world. The second benefit which is offered by NIAM is the grouping algorithm which enables to pass from a conceptual schema to a set of relations in 5th Normal Form.

The JSD method enables to describe the model behaviour. All entities that belong to the real world are described in terms of actions which are executed or submitted. We must declare not only the types of events which may occur to an object but also the order in which these events must occur.

6. Architecture and functionalities of the tool:

6.1. Objectives of the tool:

The objective of the tool is to introduce an aid to the specification and the conception of information systems. It is adapted as a combination of the methods NIAM and JSD which enable to represent both the static and the dynamic aspects of information systems. This approach has the following objectives:

1- To guide the designer to elaborate his conceptual schema so that a comprehensive and readable product is presented.

2- To help the designer to complete the specification of data by those of processing and behaviour.

3- To help the designer to validate his specification

4- To identify, for each phase of conception, the mechanisms which are being at work and provide text generating explanation on these mechanisms. The role of explanation is to inform and teach the user certain techniques of conception and enable him to validate his schema.

6.2. Global Architecture of the Tool:

The Tool is composed of five modules (Fig 6.1):

- The module M1: the CSDP (Conceptual Schema Design Procedure)
- The module M2: Graphic Interface NIAM
- The module M3: Graphic Interface JSD
- The module M4: The text generation
- The module M5: Specification in rewriting logic.

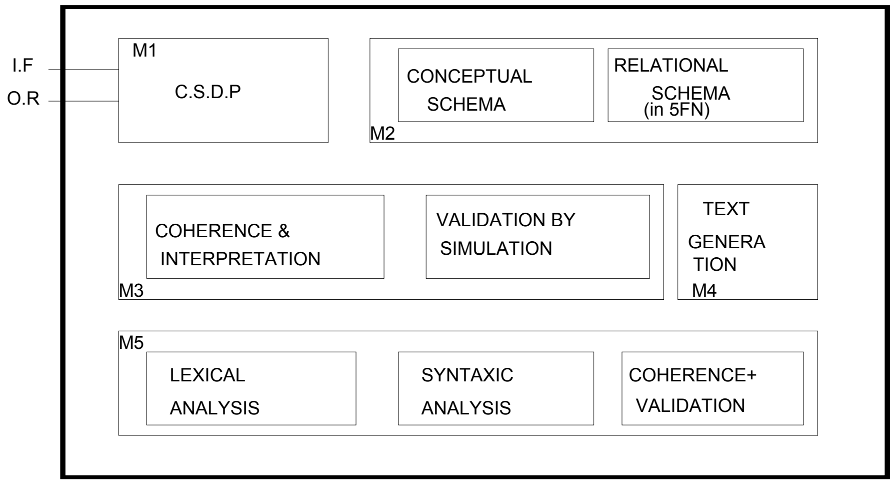

Figure 6.1: Components of the Tool

-The Module M1: It has for objective to automate the C.S.D.P. (Nijssen 89) which has been developed originally by G. Nijssen and E.D. Falkenberg. The role is to model all the phases of elaboration of a CS (Conceptual Schema) on the basis of NIAM, by starting from the Input Form (**IF**) and/or Output Report (**OR**) by passing through 9 stages to obtain the global C.S corresponding to the universe of discourse.

1- Transform the information examples into familiar elementary ideas and apply the quality verifications.
2- Draw a first draft of the C.S diagram and apply a population verification .
3- Cancel the type of identities which are surplus and the common roles, and identify the type of derived ideas.
4- Add the unicity constraints for each type idea.

5- Verify that all the types of ideas have a correct order.
6- Add the constraints of identity, totality, types and of sub-types and the occurrence frequency.
7- Verify that each entity may be identified.
8- Add the equality constraints, exclusion, sub-sets and other constraints.
9- Verify that the C.S is coherent with the original examples, has no redundancy and is complete.

- The Module M2: The static aspect is realised by this module Graphic Interface NIAM (Abdi 93) having for objective to grasp a NIAM schema with possibility to update, to analyse its coherence with respect to NIAM rules and provide the relational schema corresponding to 5th Normal form (application of the grouping algorithm). Figure 6.2 shows the NIAM schema for the case of the library taken from (Habrias 92). The object NOLOT "book" referenced by its ISBN number such that its copy is a document identified by an inventory number. This latter is delivered due to a command which is identified by a command number. It is the book which makes the object of command and this latter is delivered at a certain date.

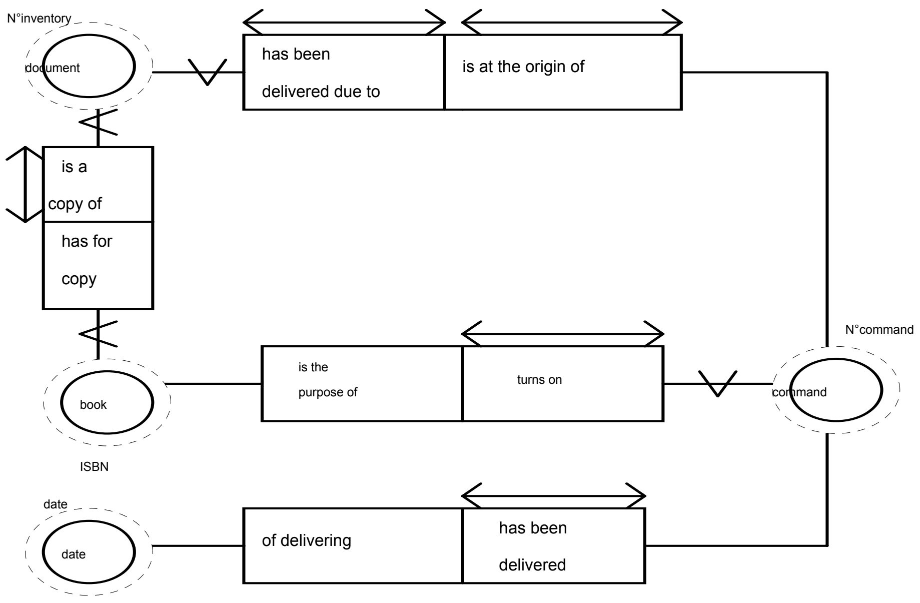

Figure 6.2: Example of NIAM Schema.

The application of the grouping algorithm on NIAM schema (above) gives the relational schema in 5th normal form (without vanishing values):

Document (<u>N° inventory,</u> N° command, ISBN)

Book (ISBN)
Command (N° command ISBN)
Delivered Command (N° command, date of delivering)
Calendar (date)

The Module M3: The dynamic aspect (processing + behaviour) is realised by this module Graphic Interface JSD (Bounadja 93). It has for objectives to grasp a CS (based upon JSD) in interactive or graphic manner with the possibility to update it , to analyse its coherence with respect to rules of JSD and provide the interpretations of specified schema (which are structured texts and regular expressions to have more semantic).
Besides, in ordert to have concretely the behaviour of the system (transition from a state to an other in front of the events of the external world). We have simulated two case studies: "Telephonic exchange" (Jean 89) and "The case of the Library" (Habrias 92).

The figure 6.3 illustrates the life diagram of the "book" entity in JSD.

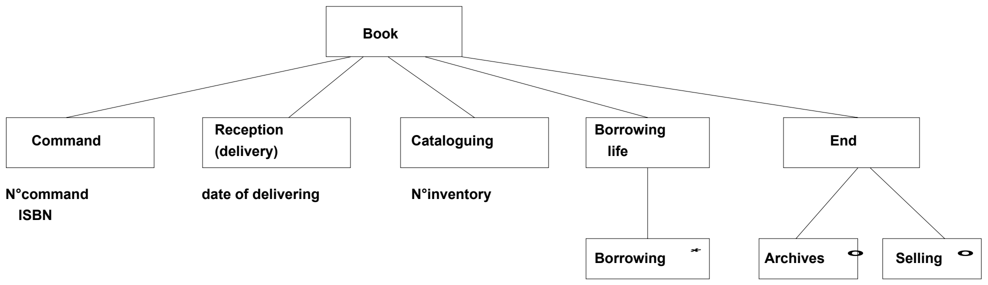

Figure 6.3: Example of JSD Schema.

The semantic of the "Book" entity is completely described. The book is referenced by its N° ISBN, commanded by the library and received after a delivery at a certain date, will be catalogued and attributed as N° inventory and can be borrowed many times to be finally archived or sold.
The Module M4: This module enables the user to validate easily his schema with the inputs of module M1 (input form and output report). This consists of paraphrasing the C.S, that is to express by a set of sentences the conceptual definition of information system such as it has been constructed by this module.
The process starts from the representation of schema saved in the data base in the form of LOT, NOLOT, BRIDGE and IDEA , to provide a text in natural language in the form of simple understandable sentences by the user.
Explanation types:
The explanation types can be classified by an elementary form or by a composed one.
(1). An elementary form is defined by two NOLOTs linked by an IDEA and possibly by its cardinalities.
To provide an elementary explanation revert to clarify the first role between the two NOLOTs and the second role (inverse role)

A <NOLOT1 > <Role1> <NOLOT2>
A <NOLOT2> <Role2> <NOLOT1>

Example: The elementary explanation corresponding to the idea between Book and Command of the figure 6.4.

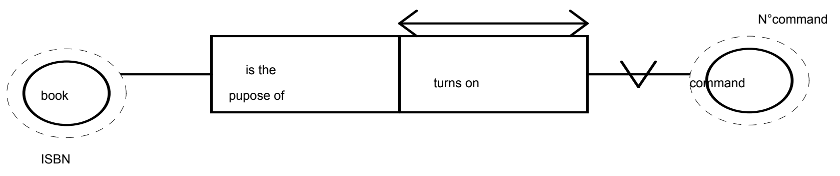

Figure 6.4 Example

The explanation for the first role is: ***A Book is the purpose of Command*** .
The explanation of second role is: ***A Command turns on Book***.

(2) The composed explanation is the group formed by an object and all the other objects directly linked to it by associations (Bridge/Idea). It consists of expressing for a given NOLOT its hierarchy and its structure according to a strategy. The adopted strategy allows to determine, departing from a NOLOT its identifier, the Bridges and the Ideas to which it takes part.

In this way, the composed explanation for the object Book of the figure 6.4 will be:

Book is a NOLOT
Book is identified by ISBN
Book for not having Document
Document is a copy of a single Book
Each Document is a copy of a single Book
Book is the purpose of Command
Each Command turns on a single Book.

The Module M5: This module allows to check in a strict way (automatically) the specification done by the M3 module. To manage it , this specification have to be, according to us, transformed to the formalism of the rewriting logic in order to be grasped, lexically and syntactically corrected and later validated (checking the properties of the specification: completness, consistence...) by this module. Its role is to satisfy and to refine more and more the third objective of the tool (see 6.1). It means that, in addition to the validation done by the M4 module (Text Generation) concerning the static aspect and the one done by the M3 module (Graphic Interface JSD) concerning the dynamic aspect (by simulation), this M5 module validates, in a formal and strict manner, the specification of the system as well as its behaviour in time. For example, consider the following rules infered from the case study (Auto-exchanging telephone):

rule1:(begin_ring)<tl:telephone/state:free,number:N, ring:inactive ==>
tl:telephone/state:busy, number:N, ring:active>(OK)(connection);

rule2:(begin_ring)<tl:telephone/state:busy,number:asked_Number, ring:active==><tl:telephone/state:busy,number:Number_called,ring: active>(the called_busy);
rule3:(end_ring)<tl:telephone/state:free,number:N,ring:inactive==> <tl:téléphone/state:free,number:N,ring:active>(give_back_ equipment);

7. CONCLUSION:

We have introduced the characteristics of a helping tool to the specification of computer systems having as objectives, on one hand to take into account static and dynamic aspects (of the Information System) presenting to the user a product which is readable and easy to be understood (CS), and on the other hand to help the designer during the conception process and especially to be able to validate his specification in order to be sure of the latter before going to the implementation phase.
To do that, a combination of the NIAM, JSD methods and the rewriting logic formalism seems necessary in our work, in order to satisfy the objectives of our tool. Finally, we think that the satisfaction of both the user and the designer is an ideal objective (especially for information systems), but its implementation is not easy to be reached because according to us, it tries to amalgamate the advantages of the informal and those of the formal ones , but the transition from a specification done by an informal method to its equivalent.

REFERENCES

1. Abdi, M.K. T.Taîbi, T.Nouari (1993). Conception d'une interface graphique pour la saisie et le regroupement des SC a base de NIAM, Mémoire de fin d'études d'ingénieur d'état, University d'Oran , Septembre.
2. Bodart, F. Y. Pigneur (1989). A model and a language for functional specification and evaluation of information system dynamics ,in Formal models and practical tools for information system design, Schneider (ed), North Holland.
3. Bounadja, A. F.Meghdir, M.K. Abdi (1993). Réalisation d'une interface graphique pour la saisie des S.C. à base de JSD" Mémoire de fin d'études d'ingénieur d'état . University d'Oran, Septembre.
4. Brodie, P.C. (1992). Active and Passive Components modeling ACM/PCM in (Lundeberg 92).
5. Bubenko, K. (1992). A declarative approach to conceptual information modeling in (Lundeberg 92)
6. Cammeron, J.R (1993). JSP and JSD The Jackson approch to Software Development IEEE Computer society mg.
7. Chan,E.F, F.Hlochovsky (1990). A graphical database design aid using the E/R model, in proc of int. conf on ERAy, chen (ed) North Holland.
8. Chen, F. (1976). The entity-relationship model : toward a unified view of data, ACM Tods vol1.
9. Corig, (1971). Conception d'un plan directeur d'automatisation, Documentation technique CGI.
10. Dahl, R. D.Erikson, L.A Johanson (1995). Ramatic : A modeling Support Systems SYSLAB report nb 34, July.

11. Dzenan, R. D.Pascot, P.Legendre (1988). Une approche de developpement rapide centree sur le modele logique de donnees: Les logiciels CONSOI, pp. 435-460. Congres Inforsid, La rochelle 8-9 July.
12. Gane, C. T. Sarson (1989). Structured systems analysis, Prentice Hall, Englewood Cliffs.
13. Habrias, H. (1992). Le modèle relationnel binaire Méthode Ia (NIAM), editon Eyrolles.
14. Jean, Marie Filloque (1989). La méthode JACKSON (JSD) - Laboratoire d'informatique de brest - Actes des 3émes journés, pratique des méthodes et outils IUT. Nantes.
15. KounineF, B, Abdi M.K, Rahmouni M.K (1994). A Help Tool for Information Systems Design- A Contribution to the Dynamicity in actes of CARI' 94, as a poster, Ouagadougou, Burkina Fasso, 12-18 Octobre.
16. KounineF, B, Abdi M.K, Rahmouni M.K. (1995). Information Systems Design: A contribution to the Dynamicity, in Proceedings of International Conference Engineering and Production Management, IEPM'95 Marrakech, April.
17. Kouninef, B.(1997). Arabic Interface for Information System Design Third International Conference On Modelling and Simulation, MS'97 Victoria University of Technology, Australia 29-31 October.
18. Kouninef, B. T. Khammaci (1995). A formal Approach to define Abstract Database which Supports Software Environment Objects, Journal of Computing and Information , July. Peterborough, Ontario, Canada.
19. Lundeberg, M. (1992). Information systems design methodologies : a comparative review, IFIP WG8.1 working conference.
20. Mylopoulos, W. (1982). some features of the TAXIS model, Proc, 6th Int conf. On Vldb.
21. Nijssen, G.M., T.A., Halpin (1989). Conceptual Schema and Relational Database Design Prentice Hall.
22. Nijssen, G.M., T.A., Halpin; (1992). Conceptual Schema and Relational Database Design - A fact oriented approach, Prentice Hall.
23. PerneY, J. (1988). Excelerator: Reference/User guide , index Technology Corporation, Cambridge, Mass.
24. Rolland, C. C. Richard (1991). The Remora methodology for information systems design and management, in proc. of the 7th Int. Conference on Vldb, Cannes.
25. Ross, D. (1987). Structured analysis: a language for communicating ideas. IEEE Transc. On Software Engineering,, vol 3.
26. Tamassia R, Batini C. (1993). An Algorithm for Automatique layout of E/R Diagrams, Proc of E/R Approach to software Engineering, Davis ed, Elsevier sce Publ, North Holland.
27. H.Tardieu, A.Rochefeld, R.Colleti (1986). La méthode Merise, Edition d'organisation, Paris.
28. Walfard, M (1988). Precise PC-IAST: un outil associé à la méthode NIAM, in proc. Liana, Nantes, France, Sept.

تصميم نظم المعلومات في ظل المتطلبات الديناميكية

بلقـاسم كـونينـاف

جـامعة الملك فيصل
المملكة العربية السعودية

الملخص

لقد ظهرت العديد من نماذج الدلالات المختلفة التي تأخذ بعـين الاعتبـار الجوانـب الديناميكية لنظم المعلومات. ولقد ساعدت هذه النماذج لتطوير نسق المفاهيم وذلـك عن طريق تمثيل سلوك النظام الإضافة إلى الهيكلة ومن هذه النماذج على سبيل المثال لا الحصر . ACM/PCM, REMORA, DADES, IDA . إلا أن الأدوات التي توفر جميع جوانب بتصميم نظم المعلومات معدودة. يهدف هذا البحث إلى عرض أداة تشـكل آلية مساعدة تجمع بين نموذج NIAM ونموذج JSD بدف تغطية ليس فقط الجوانب الثابتة بل كذلك الجوانب الديناميكية للنظام. ولعل من أهم ميزات هذا البحـث هـو تقديم نموذج يكون فيه تصديق المواصفات بطريقة آلية أثناء العملية باستعمال منهجية Rewrite Logic

الكلمات الدالة :

صميم نظم المعلومات، نسق المفـاهيم، الجانـب الثابـت، الجانـب الـديناميكي. NIAM,JSD