51

O/S.M: An object-oriented approach for software requirements/analysis phase

Louis LASOUDRIS
Institut National des Télécommunications
9, rue charles Fourier - 91011 EVRY - FRANCE
Tel: +33 (1) 60 76 47 44 Fax: +33 (1) 60.76 44 93
E-mail: lasoudri@galaxie.int-evry.fr
Stephen LIAO
City Polytechnic of Hong-Kong
83 Tat Chee Avenue Kowloon HONG KONG
Tel 852 788 7552 Fax 852 78886
E-mail: stephen@hp9000.is.cphk.hk

Abstract

This paper presents Object/System.Modelling (O/S.M) an object-oriented method. Its foundations are based on Object-oriented technologies and on systemic sciences. As others methods, O/S.M has a lot of models for information, behavioural and functional structure, and suggests a guideline for analysis and requirements phases. The use of O/S.M leads analysts to capture requirements and to validate their results.

Keywords

Object oriented methods, system science, modelling technics, information system, object-oriented database.

1 INTRODUCTION

Object/System Modeling (O/S.M: [Liao, 1993], [Lasoudris, 1993]) is an object-oriented method for software requirements/analysis. This method based on the system theory [Von Bertalanffy, 1948] and software object-oriented approaches [Rumbaugh, 1991], [Schlaer 91],[Jacobson, 1990], [Coad&Yourdon90], [Desfray, 1991] uses an object-oriented modeling technics and a guideline for requirements/analysis phase using system theory [Simon, 1962]. The model considers the system to study as an integration of several entities perceptible in real world with their own finalities. The interaction between these entities allows them to collaborate for realizing the system finalities.

Q. Sun et al. (eds.), *Computer Applications in Production Engineering*

The guideline combines the ascending approach (Increasing the level of complexity) and the descending approach (Refinement as in classical methods) [Ross, 1977] permitting respectively first the complexification of the system using conjunction and secondly the simplification using disjunction. Nevertheless, O/S.M gives us a mean to verify the two approaches in a set of coherent and integrated schematas (Information, dynamic, behavioural, functional) which are closed with information model, states model and data flows diagrams as they are defined in O.M.T [Rumbaugh, 1991].
Main software methods use the entity-relationship metamodel [Chen, 1976] for expressing the data semantics, and the Data Flows Diagrams [Gane, 1979] for expressing respectively the data and process (also flow controls) on these datas. The E-R metamodel and D.F.D metamodel will be often respectively implemented by the relational model and by procedural languages at design and implementation levels. These methods, for example SADT [Ross, 1977], have been greatly succeeding since 1980s. Nevertheless today, face to the complexity of the applications and the evolution of the technologies, it is necessary to take into account new software analysis and design methods [Booch, 1991].
System science [Von Bertallanfy, 1948] is a thought method which leads to solve complex problems. This science proves its efficacity in several domains as ecology, medecine, urbanisation, factories organisation,... Since Descartes [Descartes, 1642], we use a refinement thinking mecanism to solve complex problem We have to divide a complex problem to solve it. In software life cycle, requirements/analysis corresponds to the problem space [Booch, 1991]. Most software problems studied are complex.
The bases of Systemic lies on one main idea. All things in real world are complex to understand. So it is not sufficient to divide a problem to understand and to solve it. On the contrary we have to complexify it to understand this problem. System Science brings some radically different ideas. So we have to think about many problems in terms of complexity (Le Moigne, 1990). Software problems are very complex. Then it seems us that it is possible to bring closer both systemic approach and refinement approach with the goal to increase efficiency in software requirements/analysis tasks.

2 STATE OF ART

The object-oriented approach seems capable to bring some solutions face to the complexity for the applications and the evolution of the technologies.
The first tendency is towards the improvement of the modeling power of the existing models. For example, an E-R model extension ERC+ [Spaccapietra, 1989]. Extensions to E-R model allow us to model with new concepts like generalization/specialization hierarchy and aggregation hierarchy. Nevertheless, encapsulation concept is not proposed.
The second tendency is to propose the object-oriented models for the data bases designs. For example O.R.M. [Pernici, 1990]. Powerful concepts are used like encapsulation which lead us to model static and behaviour structure object. The concept of "inheritance", allowing shared data and methods between class, is one of the more important elements in these models.
Others models like state diagrams increase our thinking about software problem space. For example, state diagrams realized object by object increase our thinking. We are more precise and it is an easier way than the classical way of thought.
An other way to increase thought is to include the environment of the system, to indicate why such event, such message has arrived, and to study the retroaction from the environment to the data base system. Nowadays O.O.S.E [Jacobson, 1990] increases on users cases. With an efficiency manner, Jacobson integrates users needs in requirements/analysis process.

3 O/S.M: OBJECT/SYSTEM MODELLING

Object/System Modelling is an object-oriented method which help analysts in software requirements/analysis tasks. O/S.M proposes a guideline to map the analysis results to object-oriented database O2 [Bancilhon]. O/S.M. based on the system theory and on the object-oriented paradigm, increases the semantic model power and the thought process power. O/S.M model improves the user interface study and the user restitution of the results. Based on these two paradigms O/S.M ensures an homogeneity in the creative thought for understanding and designing a new software analysis.

3.1 O/S.M model.

O/S.M. considers that a real system is composed by a set of entities. An entity takes part in the activities of the system either as an activity executor or being useful for the activity. It activity takes part in the realization of one or several finalities of the system. The evolution of an entity is represented in a space, time, form reference system. An entity exists through the time and through the space. It travels through the time and the space with modifying its form. An entity is associated to a family which determines its structure and behaviour and also its finalities.

Family definition determines constraints on properties, on methods and also on events which could occur in the entity life.

An other originality of O/S.M is an enlarged encapsulation concept. In fact, for O/S.M an entity of a family encapsulates his own entirely history (set of events which occur in the life of the entity). O/S.M defines also constraints on events arrival. A language and a tool built with Graphtalk [Parralax, 1990] is useful to describe constraints, sequence of events. The internal part of entity defined by Family structure and valued for each entity is its characteristic.

Entity has an external part named context. All entities which interfere with a particular entity build up the context of this particular entity. The interference between entities may be static or dynamic.

Static links are classical links like genralization/specialisation, aggregation/part-of, association. Dynamic links are based on shared events, generated events, or characteristic events.

O/S.M increases study of collaboration between entities using the concept of "process of collaboration", which looks like schemas in [Desfray, 1991] or subject in [Coad, 1990], or modules in [Rumbaugh,1991]. This concept allows an analyst to use an ascending approach closely with System science paradigm. After a local study of family object, O/S.M leads to a global analysis by complexification, regrouping family objects in a "process of collaboration" families. Process collaboration is a good mean to look for families in O/S.M process.

O/S.M. classifies the families into three types: the family of agents, the family of passive objects and the family of time (clock, calendar,..).

An agent is an intelligent entity which is able to take decisions. It can initiate at least one "process of collaboration" for reaching the finalities of the system.

A passive object is not an intelligent entity. It undergoes the transformation in the reference system (space, time, form) and is used by the process of the system to study. A passive object can not make any decision, but it propagates the character or context modifications toward the other entities which possess a dynamic link with it.

Entity of time family takes part in the entirely process of collaboration. Its context is composed by all the entities of the real system.

Internal functioning of an entity

A family defines a behaviour for its entities. A function is triggered by event which is produces by the entity itself or by the entity context. So, one or several functions are executed. the result of processing function are one or several events which impact the object itself (we name this event: characteristic event) or its context (by the means of shared events or generated events).

Global functioning of system

When an event has been noted on an entity of a family, it can bring:

- •the generation of a other event on one or several other entities.
- •the note of the same event on one or several other entities.

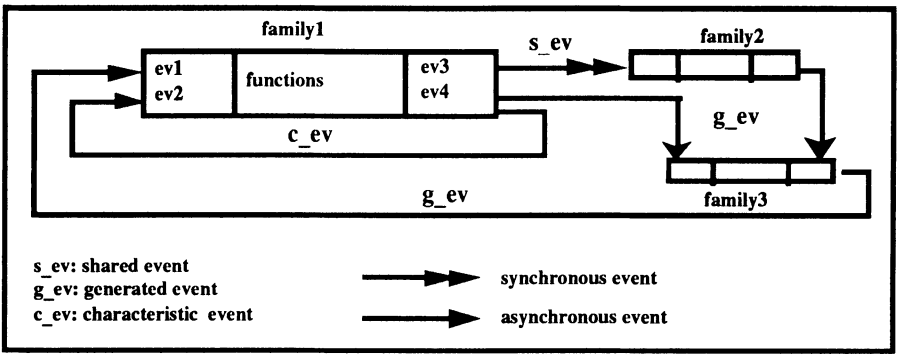

Figure 1 Global and internal functionning of Object/System Modelling.

According to the system theory, a family is a system including a set of entities. It encapsulates a system, a family is itself an entity having properties, functions, events and a context (static links, dynamic links, process of collaboration). This entity can be dynamically bound with other entities through shared or generated events.
According to the system theory, actor is a system This concept permits to take into account organizational aspect of the software system. The actors are the entities of a families of agents. A family of actors is characterized by a name, the finalities, the functions, the events and the knowledge. So we could modelize organization aspects then increasing understanding of software environment or system environment.

Family Models

The first model corresponds to **static structure of family** (Figure 2). For each family we choose to represent Family linked with its roles and attributes. Links between concept are typed.A circle represents a Component Family. A square represents a Family Role. > is the monovaluation symbol while >> is multivaluation symbol. Dotted line represents optional link while full line represents Mandatory link. The couple (x,y) represents for x = F Fixed value, x= L for Free value. For y= P permanent value and y = T Temporary value.

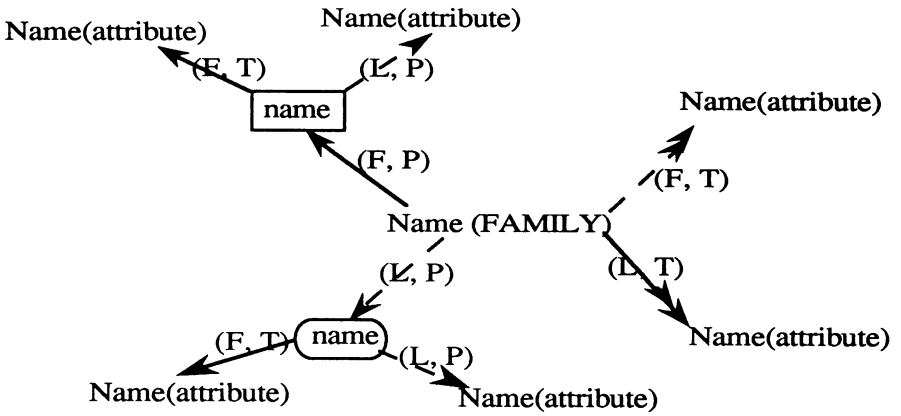

Figure 2 According with the above text representation of static structure of a Family.

Dynamic aspects are represented by state diagram. O/S.M emphasises the concept of role of entity. State entity and role change are represented with the same model.

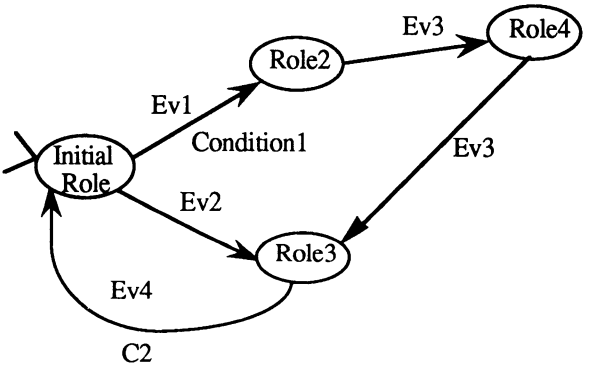

Figure 3 Links representation between roles. We could notice that states of entities are representing with the same model.

Interaction between Family

Static interaction model represents families of the system and links between these families. Monovalued situation link is represented by a full line and arrow, double arrow if the link is multivalued, dotted line if the link is an existence link. The couple (x,y) represnts for x = T Total link x= P partial link y = T Temporary P permanent then we obtain:

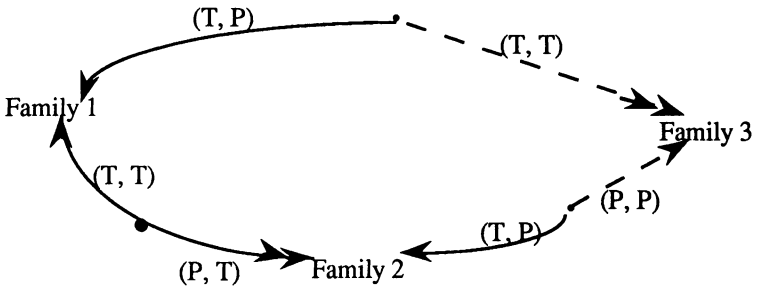

Figure 4: Links Representation between Families.

Dynamic links between entities.
This model presents for a process of collaboration the events sequence.

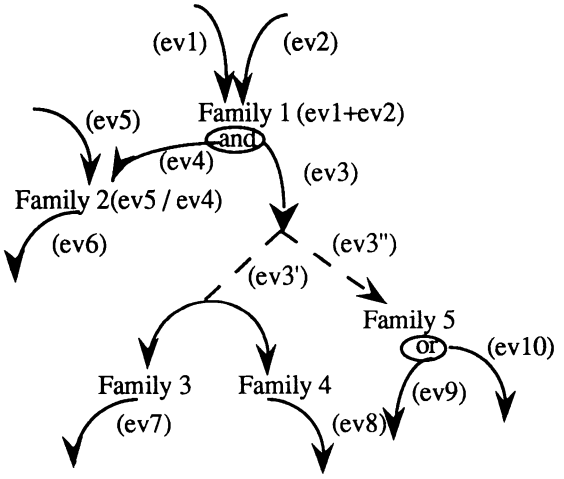

Figure 5: Dynamic links between Families.

3.2 The Requirements/analysis modelling guideline

O/S.M. proposes a modeling guideline permitting analyst to understand and to model a software system. O/S.M proposes a set of advices to lead requirements and analysis phasis. It proposes interviews and validation of interviews using the models. O/S.M uses the concept of perspective to classify the entities, which permits us to get more homogeneous subset. Face to the life cycle of software, O/S.M. brings a help for the comprehension and the representation of the existing and the future software. The guideline comprises four modelling stages, the descriptive stage, the analysis stage and validating stage and mapping stage. Note that the process is a non linear process. We have to repeat each step along the analysis and then we have to build up an application progessively.

The descriptive stage

•Determination of the existing system.

Using process of collaboration, the analyst is able to delimit the space problem. It is when context event associated with process of collaboration has no impact on others non described entities.

•Emergence of the families corresponding to collaboration process.

In an analysis phase it is possible to start with descending decomposition when we meet large system or when there are a lot of families.

•Determination of the fundamental families.

In this step we look for families by grouping entities with similarities in information, behavioural structure.

•Opened prototypes of the fundamental families.

The descriptive stage ends with a gross description of future families. Each families is associated with a prototype which describes static and dynamic structure. At the end of this stage, each entities of real world is instance of one family. Each family describes the attributes and the functions of its entities. The events which could occur on entities of this family are described with their triggers and predicates.

The analysis stage

•Determination of new families by disjunction.

In this step we refine fundamental families using subset theory. So it is a good way to make emergence of new families which are more precise and more homogeneous.

•Determination of new families by conjunction.

It is also possible to obtain new families using fundamental families collected in the first stage. Using conjonction principle we group any fundamental families which particpate in a process of collaboration. In fact we could use this conjunction in complemntarity with disjunction mechanism.

•Construction of the model schemes.

In this step we draw schemes named family information model, behavioural model based on state diagram which apply on entity life and also on role played by the entity. This is very useful to study mechanism of evolution of instanciation link between family and entity.

•Restricted prototypes of the final families.

In this step, we refine constraints on attribute. We specify functions with more details.

We complete also model by additionning some details.

The validating stage

•Application of the validating rules.

O/S.M suggests some rules to be applied on model, or on coherence between models. For example function encapsulated in a family is associated with states of entity of this family. In each state, we have to trigger a function and this function must be declared in final families.

The mapping stage

•Rules to map O/S.M to O2 database.

O2 is an object-oriented database system [Bancilhon,1990]. O/S.M suggests some rules to map attributes and functions of each family in D.D.L and methods described in CO2 language.

4 CONCLUSION

In this paper we present an object-oriented method for analysis software. we use models and guideline suggests by O/S.M in bank application. Using this methods leads us to modelize some applications. nevertheless we have to demonstrate that it is able to be used in others contexts like industrial context. We define also a CASE for this method using Graphtalk metatool.

5 REFERENCES

Bancilhon (1990) The O2 book. Industrial ministry 1990.
Booch (1991) "Object-Oriented Design with application"Cummings Publishing Cy
Bertalanffy (1948) System general theory.
Chen (1976) "The entity-Relationship model. toward a unified View of data"
ACM Transaction Database system Vol 1, N°1, Mars 1976
Coad(1991) "Object-Oriented Analysis" Prentice-Hall International1991
Descartes(1642) Discours de la méthode
Desfray(1991)Le modèle classe-relation (In French) application C++Masson 1991
Gane (1979) Structured Analysis and Structured Design Gane/Sarson Prentice Hall
Jacobson(1990) Object Oriented Software Engineering Addison Wesley 1991
Lasoudris(1993) Object/System Modelling L.Lasoudris and S.Liao I.E.E.E international conference Le Touquet France octobre 93.
Liao(1993)Object/system Modelling Thesis july 93 University of Aix/Marseille 3 France Pr J.L.Le Moigne
Le Moigne (1990) "La modélisation des systèmes complexes" DUNOD 1990 (In French)
Parralax (1990) Graphtalk a CASE tool for metamodelization.
Pernici (1990) "Requirements specifications for object-oriented systems" INFORSID 90 Nouvelles perspectives des systèmes d'information EYROLLES
Rumbaugh (1991) Object modeling technicsPrentice Hall
Ross(1977) "Structures Analysis for requirements definition" IEEE Trans. Software Eng. Vol SE-3, No1
Schlaer(1991) Object-Oriented systems analysis modeling the world with Data. Yourdon Press
Sibertin-Blanc(1991) "Cooperative objects for the conceptual modelling of organizational information systems"IFIP TC8/WG8.1 working conference on the objected-oriented approach in information systems Quebec Canada .
Simon(1962) The architecture of complexity Philosophy of sciences
Spaccapietra (1989)"About entities, complex objects and object-oriented data models"Proceedings of the working conference on information system concepts Namur, Belgium

5 BIOGRAPHY

Louis LASOUDRIS is an information system teacher in Telecommunications National Institute (I.N.T), engineering school of France Telecoms in Evry France. His main interest is Object oriented methodology and more precisely Modelling technics. His research leads him to suggest a new object-oriented method O/S.M. Now, he works on a CASE tool using Case Based Reasoning technologies to allow analysts reusing models.

Stephen LIAO was a student during three years in I.N.T. He worked on a thesis on object oriented method with Pr J.L.Le Moigne. He creates with Louis Lasoudris O/S.M. Now he is a teacher in polytechnic school of Hong-Kong, where he teachs Object-oriented database, language and methods.