OLSEN: An Object-Oriented Formalism for Information and Decision System Design

Ramzi Guetari, Frédéric Piard[1], Bettina Schweyer[2]

LLP/CESALP 41 Avenue de la Plaine
BP. 806 - 74016 Annecy Cedex - FRANCE
Tel : (+33) 50.66.60.80 - Fax : (+33) 50.66.60.20
email : guetari|piard|schweyer@esia.univ-savoie.fr
[1]CIFRE contract with ANRT and Pôle Productique Rhône-Alpes
[2]CIFRE contract with ANRT and ARM Conseil

1. Introduction

The Object oriented model has spread widely within programming languages during the last years. The principles of this model have had a great influence on analysis and design techniques. However no existing method is able to manage the whole analysis-specification-design-implementation cycle, preserving the homogeneity of the model used in different stages and the coherence by passing from one stage to the following.

We think that the global management of the life cycle cannot be solved, with the existing state of knowledge, by one unique miraculous method, which could adapt to every kind of application. We think on the contrary that the problem should be treated by a panel of methods dedicated to a particular domain.

For this reason we have developed the OLYMPIOS model at the LLP-CESALP laboratory. This model covers the life cycle of every application in the field of Information and Decision Systems for Manufacturing Firms. OLYMPIOS uses algebraic techniques, transformation rules and a predefined entity organisation to propose an original approach for object oriented design of information and decision system.

2. OLYMPIOS Model Concepts.

The information processed in an enterprise, which we call industrial information, is a complex datum. An information and decision system (IDS) must take this complexity into account. We propose to represent industrial information through four main facets :

- data, describing the different entities handled by the IDS and the actions that they can perform or can be subjected to ;
- temporal properties of the different kinds of processes (including traceability of information) ;
- organisation, considered through information flows;
- economic facet, which describes the means of performance evaluation in relation to enterprise environment and objectives.

The OLYMPIOS model [Beauchêne 93] [BHP 93] [BHS 93] covers the different stages of such a system life cycle and proposes original solutions for its analysis, specification, design and realisation. OLYMPIOS describes activities, taking into account the assigned objectives and the resources availability. The basic modelling elements are :

- an industrial information database, where products, resources, machines,… are described.
- Consumer-Supplier Information Systems (**CSIS**). A CSIS stands for an “atom” of organisation. It is a generalisation of the customer-supplier exchange relationship to every couple of actors in the enterprise (men, machines, software). Every CSIS is associated to an objective, transforms resources and emits a satisfaction level.
- an Objective Management System (**OMS**), whose role is to create a graph from expressed objectives, where every node is an objective associated to a CSIS.
- a Resource Management System (**RMS**), in charge of the product and resource management and sharing.
- an activation system (**AS**), producing actions plans to organise processes, taking into account the application, temporal constraints, and communications/synchronisation between CSIS.

3. The IDS Life Cycle

The OLYMPIOS model covers the different stages of the IDS life-cycle (Fig. 1). We use an algebraic approach for the four facets of industrial information so as to obtain a coherent (i.e. sufficiently complete and consistent) specification. The design stage enables us to design the information system from specification and by analysing the "existing" system of the enterprise and its objectives. The result of this stage is a representation of the IDS using structured entities. The OLYMPIOS model introduces the uniformity of the model used from specification up to design. It uses tools proving the coherence of the system in the specification step and maintaining this coherence by automating the translation from one stage to another.

3.1. Analysis Stage

In the analysis stage, the relevant information for the data, the temporal, the organisational and the economic facets is collected.

The result of the **data** facet analysis consists in the description of the data handled (resources etc.) in the system to design and, for each datum, the set of operations that can be realised (data dictionary). This static description can be translated into a finite state automaton in which every node represents a state of the datum in question and every edge an operation which produces a new state.

The analysis of the **organisational** aspects of the manufacturing firm results in a set of interactions between the different agents of the enterprise in the form of exchange relationships. By interviewing each of these agents we enumerate, on the one hand, the exchange relationships in which he is consumer, i.e. follows a certain objective by asking for satisfaction of the respective needs, and on the other hand, we identify the relationships in which he is supplier and performs a certain function. For each of these functions (which we would like to call basic operation) he enumerates the resources necessary for realising this operation and the algorithm he follows to obtain the wanted resource.
Thus, this interview gives us information about

- objectives and their decomposition,
- identification of the possible suppliers for the realisation of a given objective,
- the basic operations that can be performed and knowledge about how to execute the operations and which resources are needed.

Starting from this information, we can establish a knowledge base of the different ways to decompose objectives and a knowledge base for the needed resources for each basic operation. These knowledge bases will help us, in addition to the predefined structure of such an exchange relationship, to define the enterprise organisation.

The analysis of the **temporal** facet provides a dynamic description of the system. It enables us to describe the temporal behaviour of different agents and resources of the system and their interactions. For this part of the analysis, a method close to natural language is being developed which will allow a user-friendly way of describing temporal rules.
From this analysis we also obtain a description which we call *realization programs*. These programs contain the description of the CSIS functionning and of the operations which are not formally describable.

As far as the **economic** facet is concerned, we are actually working on an interview structure including fuzzy logic in order to acquire the information necessary for evaluating the system's performance.

3.2. Specification Stage

3.2.1. Data Specification

The data facet corresponds to the IDS functional and structural aspects, and aims at representing the technical and technological data. We use Algebraic Specifications of Abstract data Types (ASAT) [Guttag 78] [Jacquenet 86] [Liskov 87] so as to have efficient and simple proof techniques at our disposal. An ASAT enables us to express an entity behaviour in a high level formalism. For a given entity, an ASAT is a triple $<\Omega,\Sigma,A>$, where Ω is a set of domains containing the domain of the entity values, Σ is a set of operations on the entity, and A is a set of equations (axioms and preconditions) on these operations, which determines the entities behaviour and the relationships between them. ASAT are automatically constructed from the entities automata, which are the result of the analysis stage. This automatic construction is realised by the algorithms [Nkongo 90] developed in our laboratory.

3.2.2. Organization Specification

It starts from the analysis of the "existing system", which results (inter alia) in the identification of actors and their functions and objectives. Specifying organisation consists in formally expressing identified objectives (in the "triple" form), and in constructing their associated CSIS from standard parametrized ASAT of organisation [Beauchêne 93]. Simultaneously, one must elaborate the different graphs of objectives.

3.2.3. Temporal Specification

The specification of the industrial information temporal facet uses a synchronous process algebra, directly derived from the SCCS calculus of R. Milner [Piard 93]. We specify four kinds of processes with this language :

1- chronological and event-based clocks, essential to specify synchronisation and to measure temporal intervals;
2- behaviours of data facet entities, which are not completely determined by ASAT axioms;
3- behaviours of CSIS;
4- activation plans, elaborated by the activation system from graphs of objectives and resources to schedule the CSIS.

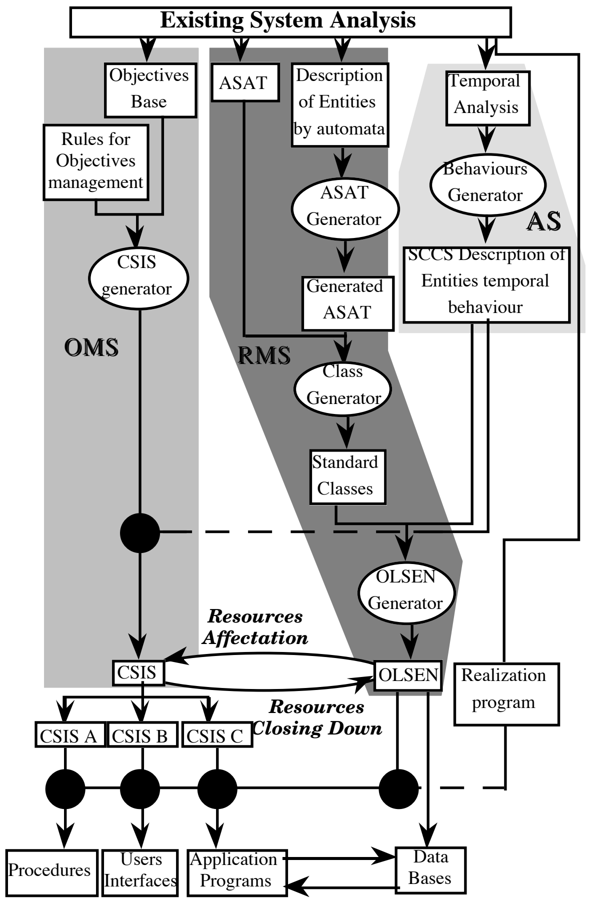

Fig. 1. The Analysis - Specification - Design Cycle in the OLYMPIOS Model

3.2.4. Economic Specification

This facet cannot be specified independently of data and organisation. Indeed it is shared between them, and the most important part is included in the organisation facet. Works are still going on to sharpen the economic view of OLYMPIOS on the information system (with the help of performance indicators, fuzzy logic and project-based management approach).

3.3. Design Stage

The OLYMPIOS model, in its design stage, is based on the class model. This model was extended in order to allow to take all industrial information features into account, in particular real time ones. The result of the design stage is an organisation of entities independent of possible target programming languages : OLSEN (**OL**ympios **S**tructured **EN**tity).

An OLSEN [Guetari 94] is composed of a "class" part and another part called "scenario" which indicates the interactions with its environment. The difference between an OLSEN and a classical object is the scenario which describes the temporal behaviour generally missing in the standard class model. The OLSEN model is a "design object".

In this paper, we present only the specification and design of Activation System (AS part) and Resource Management System (RMS). The Objective Management System is the subject of a publication to come.

4. The Transition from the Analysis to the Specification Stage

This stage consists in describing data types using finite state automata. We must first insist on the fact that every entity cannot be described by an automaton. Only if it has successive states and if it is concerned by actions passing from one state to another can it be described by an automaton. We do not use the automata as a specification tool but as a tool allowing us to shape the evolution of some kind of data type over a set of states. In this kind of automata, each transition represents an operation changing the entity's state and each node represents one state of the entity. The automata may have many transitions corresponding to the same operation, however, each state is unique. A particular state called "starting state" must always exist. It corresponds to the extremity of the transition which stands for the operation creating the type of interest (TI).

The entities described by automata are distinguishable by the successive states that they can have. The order in which different states are occupied is well defined. The graph of state changing is oriented and has a starting state from which we can observe the evolution of the entity. This graph allows us to distinguish the constructor operations using a single method. The transitions corresponding to these operations have extremity nodes which can be reached from the starting state by only one path of the graph. The construction of axioms is done in two steps : the construction of left parts of axioms and the construction of right parts of axioms, as it is shown below :

The construction of left parts of axioms :

The construction of axioms left parts consists of building the following sets :

- CT = {c(y*), c ∈ C}
- OT = {o(x, y*), o ∈ O, x ∈ CT}
- ST = {s(x, y*), s ∈ S, x ∈ CT}

OT and ST contain the left parts of specification axioms. Axioms which define the semantic of the abstract data type have their left parts in the OT set and axioms which shows the simplification of terms of $T(\Omega,\Sigma)$ have their left parts in the ST set.

The construction of right parts of axioms :

The graph of states, whose every node is a state of entities of TI type, and whose every transition is an operation, provides :

1- Ω = {TI, STATES}, STATES = {E1,E2,E3,...}
2- Σ = {state, $\sigma 1$, $\sigma 2$, $\sigma 3$, ..., σn} = O+C+S, T = S + C = {$\sigma 1$, $\sigma 2$, $\sigma 3$, ..., σn} is the set of operations which create or transform the values of TI (represented in the automata by transitions), O={state} contains a single observer.
3- Left parts of axioms by the building of AC,AO,AT from O,C et T.
4- Right parts (y) of axioms in the form *state(c(x*)) = y*, where c∈ C, and y is the expression of the name of the node extremity of the path represented by c(x*) from the starting state. If there are many of these paths then the y term will be expressed in the form **if**...**then**...**else** ...
5- Right parts (y) of axioms in the form *s(c(x*)) = y*, where s ∈ S is a convertible operation and y corresponds to the canonical form of the state extremity of the path c(x*), i.e. the expression of the shortest path between the starting state and the state extremity of the path represented by the expression c(x*). In other terms, these axioms are represented in the automata by simple circular paths. If there are many of these paths then the y term will be expressed in the form **if**...**then**...**else** ...
6- Preconditions related to the state of arguments (membership of TI) of each operation, which are expressed by the restrictions on the domain of this operation before its execution. These restrictions are issued from the state origin of the arc representing the operation.

5. The Transition from the Specification to the Design Stage

The transition from the specification stage (ASAT and SCCS) to the design stage is done automatically in two steps. The first step consists in taking the ASAT one by one and translating each one into a standard class. The second step is a global one and permits the organization of the communication between the obtained classes. The benefit of this automation is the preservation of the coherence obtained in the specification stage.

5.1. The Standard Class Generation

The class attributes and methods are generated from the ASAT operations. This is done using the following rules. We note an operation : $\sigma : \Omega_1 \rightarrow \Omega_2$. Ω_1 is the set of domains and Ω_2 is the set of codomains. "TI" is the data type that we specify. We distinguish three kinds of operations :

- Case 1 : $\sigma : \Omega_1 \rightarrow \Omega_2$ / TI $\not\in \Omega_1$ and $\Omega_2 = \{TI\}$. This kind of operation corresponds to a particular constructor. For each constructor, we generate a method "*New*" with parameters of type Ω_1.
- Case 2 : $\sigma : \Omega_1 \rightarrow \Omega_2$ / $\Omega_1 = \{TI\}$ and $\Omega_2 = \{\omega \neq TI\}$. This kind of operation corresponds to observers. The class structure is obtained from these observers. For each observer we generate an attribute of type Ω_2 and a method to access it.
- Case 3 : $\sigma : \Omega_1 \rightarrow \Omega_2$ / TI $\in \Omega_1$ and TI $\in \Omega_2$. This case corresponds to a general one. For each operation of this kind we generate a method with in parameters of type $\omega \in \Omega_1$ / $\omega \neq$ TI and out of parameters of type $\omega \in \Omega_2$ / $\omega \neq$ TI.

The scenario of an OLSEN is issued from SCCS formulae. An SCCS formula contains several deterministic parts. Each part provides one script in the OLSEN scenario. The scenario generation is done in three steps : the first two provide the declarative part of a scenario, the third one provides the dynamic part. For each OLSEN, we determine the determinist parts of the corresponding BEHAVIOUR (separated by a "sum" operator). For each part, we execute the following three steps :

- **Event Detection**. This step permits the detection and declaration of the different kinds of events. The type of each event is deduced from the SCCS syntax. A communicational event appears in at least two BEHAVIOURs, once preceded by the delay operator δ, and once without this operator. An environmental event is identified by the existence of a clock emitting this event. An event is conditional if its complementary event appears at least once in a BEHAVIOUR. When all events are declared, we proceed to the unification of the communicational events. This unification is based on the *observational equivalence* [Austry 84] and consists of giving the same name to two synchronously successive events in a SCCS formula.
- **Identification of the Set of Suppliers**. For each communicational event, we define its receiving OLSENs whose BEHAVIOURs contain this event, preceded by the delay operator δ. Any OLSEN responding to this event by applying one of its methods must be added to the suppliers list of the treated OLSEN.
- **Script Generation**. A script is generated for each determinist part. Each event described in the formula is replaced by one or several simultaneous dispatches of messages. The receivers of these messages are the suppliers defined in step 2.

6. The Transition from the Design to the Realization Stage

This transition is based on the realization programs which we have obtained in the analysis stage. The OLSEN formalism helps us to generate data bases on the realization stage. The application programs are obtained through the OLSEN, the realization programs and the CSIS organization.
If we target object-oriented data bases in the realization stage, we have to use the OLSEN and the realization programs. In this case, each class part of an OLSEN is directly translated into a data base object and the scenario part is used for the data access in the application programs. The realization programs allow us to implement the methods of the data base objects.
If the data bases are not object-oriented, only the structure of the OLSEN interferes for the realization of these data bases. In a relational data base, for example, the OLSEN structure is used for the table creation. The inheritance relationship is eliminated in these data bases and replaced by the result of merging the structures of a super-class and the sub-classes.

In the realization stage we can obtain three different types of CSIS translations: automatic CSIS where the actors perform totally automated processes, semi-automatic CSIS where one of the two actors performs an automated task or the manual CSIS where both actors perform manual tasks.

The first type of CSIS with the realization programs and the scenarii allow us to obtain the application programs. These programs will act upon the data bases with the classical operations like add, modify and delete. These interactions with the data base are performed through message sending between the data base objects in the case of an object-oriented data base or through primitives which are the result of the OLSEN behaviour in the case of non object-oriented data bases.
The semi-automatic CSIS form the interactions between a user and a process. These CSIS lead towards the implementation of user interfaces and external views which restrict the data base access according to the user's rights.
The manual CSIS finally, allow us to realize the manual procedure for which the automation would be too expensive.

7. Conclusion

The OLYMPIOS model provides the means to analyse and specify coherently an industrial information and decision system. It allows then to design the specified IDS by preserving the coherence obtained in the specification stage by using algebraic techniques. The continuity and uniformity claimed by the Olympios model is the result of two factors :

- the use of algebraic tools to specify all the components of an IDS like the data facet, the organization facet or the temporal facet,
- the use of ASAT to specify data and Objects to design them.

This care of continuity and uniformity has lead us to develop algorithms (and parts of a future CASE-Tool) to automatically generate a coherent OLSEN organisation from the analysis. Our objective is to generate a maximum of code for applications.

References

[Austry□84] AUSTRY D., BOUDOL G., Algèbres de processus et synchronisation, TCS 30(1) 1984

[Beauchêne 93] D. Beauchêne. L'information industrielle : définition et spécification. PhD thesis, University of Savoie. December 1993.

[BHP 93] D.Beauchêne, A.Haurat, F.Piard. Une méthode de spécification de l'information industrielle par types abstraits algébriques. Proceedings of ICO'93, 4-7 May 1993, Montreal Canada.

[BHS 93] D.Beauchêne, A.Haurat, B.Schweyer. Designing an information system for a manufacturing enterprise under the aspect of a CIM approach : the model OLYMPIOS. Proceedings of APMS'93, 28-30 September. 1993, Athens Greece.

[Guetari 94] R. Guetari. and F. Piard. From the Specification to the Design of an Industrial Information System: the Olympios Model. Accepted in the 1994 IEEE Conference on Systems Man and Cybernetics. San Antonio - Texas October 2 - 5 1994.

[Guttag 78] J.V. Guttag and J.J. Horning. The algebraic specification of Abstract Data Type. Acta Informatica. 1978 Vol 10. P 27-52.

[Jacquenet 86] J.P. Jacquenet, P.Lescanne. La réécriture. Techniques et Sciences Informatiques 1986. Vol 5 N° 6. p. 433-452.

[Liskov 87] B. Liskov. Data Abstraction and hierarchy. OOPSLA'87 Addendum to the proceedings. 1987.

[Nkongo 90] T. Nkongo. Spécification algébrique de types abstraits pour le modèle Olympios. DEA report Ingénierie Informatique of INSA Lyon. September 1990.

[Piard 93] F. Piard, C. Braesch - Application du calcul SCCS de Milner à la spécification de processus informationnels par types abstraits algébriques dans une entreprise manufacturière. Real Time Systems Conference, Paris 1993.