An Information System Modeling with UML2 for Reports Archiving for the Knowledge Management of a School Structure

S. Demigha

CRI (Centre de Recherche en Informatique), Université de Paris 1 Panthéon-Sorbonne, Paris, France

Abstract - *This paper describes the modeling of an Information System (IS) with UML2 of a school structure in order to implement a real Information System in the UFEC Institute in Paris (Graduate School). It presents requirements engineering able to develop the Information System. To design and implement this IS, we need to determine the different factors influencing this type of computer application. These factors determine the way data will be stored and retrieved. We use the last version of UML (Unified Modeling Language) to formalize and structure data. The UML is suited with the Object-Oriented approach and deal with a high level of abstraction of data particularly when it is about managing various and complex data. A school structure is splitting in many departments which necessitate a best organization of data and knowledge. This approach corresponds to our purposes and needs.*

Keywords: Information System, Modeling, UML2, Knowledge, Management.

1 Introduction

According to the literature [1],[2],[3],[4], an Information System (IS) is defined as a collection of hardware, software, data, people and procedures that work together to produce quality information. Silver and al, [5], defined Information Systems as follows: Information Systems are implemented within an organization for the purpose of improving the effectiveness and efficiency of that organization. Capabilities of the Information System and characteristics of the organization, its work systems, its people, and its development and implementation methodologies together determine the extent to which that purpose is achieved.

Information Systems research is generally interdisciplinary concerned with the study of the effects of Information Systems on the behavior of individuals, groups, and organizations, [6], [7]. Hevner and al, [8], categorized research in IS into two scientific paradigms including behavioral science which is to develop and verify theories that explain or predict human or organizational behavior and design science which extends the boundaries of human and organizational capabilities by creating new and innovative artifacts.

Salvatore March and Gerald Smith, [9], proposed a framework for researching different aspects of Information Technology including outputs of the research (research outputs) and activities to carry out this research (research activities). They identified research outputs as follows:

- Constructs which are concepts that form the vocabulary of a domain. They constitute a conceptualization to describe problems and specify their solutions.
- A model which is a set of propositions or statements expressing relationships among constructs.
- A method to perform a task. Methods are based on a set of underlying constructs and a representation (model) of the solution space.
- An instantiation which is the realization of an artifact in its environment.

An Information System is not only the technology an organization uses, but also the way in which the organizations interact with the technology and the way in which the technology works with the organization's business processes. Information Systems (IS) are distinct from Information Technology (IT) in that an Information System has an Information Technology component that interacts with the processes components, [10].

Information Technology departments in larger organizations tend to strongly influence Information Technology development, use, and application in the organizations, which may be a business or corporation. A series of methodologies and processes can be used in order to develop and use an Information System. Many developers have turned and used a more engineering approach such as the System Development Life Cycle which is a systematic procedure of developing an Information System through stages that occur in sequence.

A school structure can be considered as a large organization. This organization is splitting in various departments which take an important part for the organization. To organize these departments we need to develop a real Information System including all actors participating at this organization and gathering them in a single and common system.

For developing this system we have focused on the best and innovative methods, tools and techniques found in the literature based on high and efficient technologies and

adopted them to build our own IS with new purposes and propositions.

We choose a traditional cycle for developing this system by following a systematic procedure and retain the Object-Oriented, [11], approach with UML2 (Unified Modeling Language), [12], for the system modeling.

The object model allows for a higher level of abstraction when representing concepts from real word. It may represent all the complexity of a school structure without losing information and coherence. Moreover, it allows for an easy creation of new data types and a flexible scheme adapting to new emerging knowledge and the modifications of the model.

The paper is organized as follows:

- Section 2 presents the UFEC graduate school structure;
- Section 3 presents the users requirements of the IS;
- Section 4 describes the object model retained to the school structure;
- Section 5 is the conclusion with further research works in progress;

Section 2 presents the overall architecture of the UFEC school.

2 The UFEC graduate school structure

The UFEC School is splitting in 8 departments:

1. The Supervisor Department;
2. The Training Department;
3. The Administrative Department;
4. The Human Resources Department;
5. The Communication Department;
6. The Placement Department;
7. The e-learning and virtual University Department;
8. And the Security Department.

Figure 1 presents the global organization chart of the UFEC school.

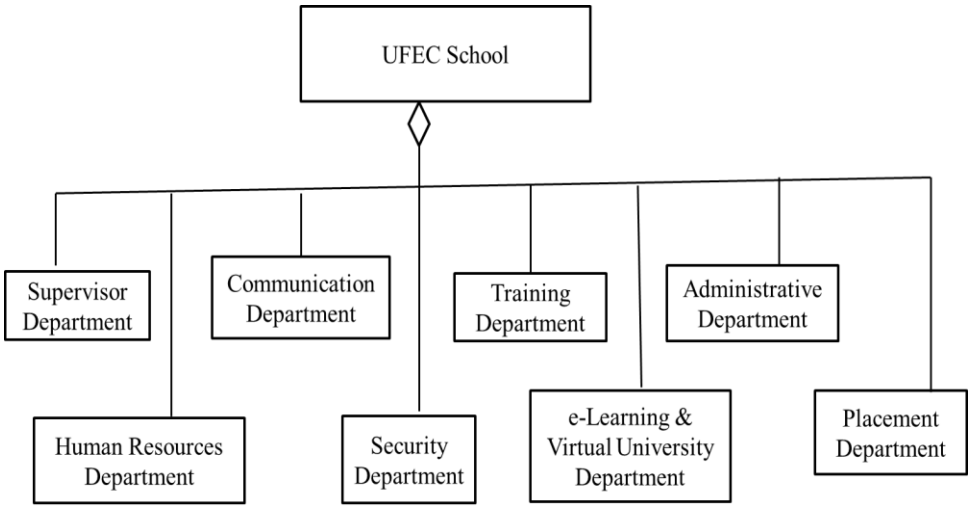

Figure 1: The general organization chart of the UFEC School

Section 3 presents the users requirements of the IS.

3 Users requirements

The development of the IS requires to set up specifications from user requirements. We have defined these specifications from the management of the UFEC school where the Information System will be implemented. We use UML to express these requirements. The UML (Unified Modeling Language) is a modeling language and has not a notion of development process, which must accompany a method. The dictionary defines a method as a systematic or orderly procedure, [13].

3.1 Staff using the IS

The public participating at the school organization belong to eight categories:

- *Supervisor staff:* aim to supervise the school.
- *Training staff:* aim to organize pedagogical contents and follow-up of students.
- *Lecturer staff:* aim to assure lessons, lead scientific works of students and evaluate them.
- *Student staff*: are concerned by the training and learning stages (Master and PhD).
- *Administrative staff:* are concerned by the administrative people (Internal and External).
- *Human Resources staff*: are advocated for both the school and the people who work in the company.
- *Communication staff:* people who disseminate information between the departments of the school and outside the school (enterprises,…).
- *Security staff:* people who control access to data; access to data must be controlled.

  Access to the system is granted either to the location of the user: outside the school (enterprises or external people) and inside the school (employees and students).

3.2 Characteristics

Data have been collected through the eight categories that we have mentioned in the previous section. These data must be analysed with UML2 and will be described in details in the following section.
Section 4 describes the Object-Oriented model retained to the school structure.

4 The object modeling with UML2

Through a model we aim to provide a better understanding of the system under development. UML2 model permits an application's design to be evaluated and critiqued before implementation, [13].
Changes are easy and less expensive to make when they are made in the early phases of the software lifecycle. Models help us capture and record our software design decisions as we progress toward an implementation. The UML2 defines a diagrammatic notation for describing the artifacts of an Object-Oriented analysis and design. We can visualize, specify, construct and document our software application. As our IS become ever larger and ever more complex we need to manage that complexity and, in a sense, simplify it which allows us to have a better understanding of it.
Finally, from our UML2 diagrams we can derive programming language code. This is referred to as forward engineering – the generation of code from UML2 models. This is an approach we advocate through this paper. The models are the core of our designs. In this paper we only focused on the modeling and the code for implementation will be programmed later in another paper. The object modeling is based on three models: the functional model, the object model and the dynamical model.

4.1 The functional model

It represents the functionality of the system and allows the modeling of users' expectations. There are two basic concepts in functional modeling: 'Users' that utilize the system, and 'Use Cases' that represent the utilization of the system by the users.

4.1.1 Users

Users are actors using the system. There are 5 types of users:

- *The lecturer:* accesses to anonymous data. Modifies only pedagogical data.
- *The student:* may access to certain type of data but has not to modify rights.
- *The supervisor:* accesses to all type of data. Can record, modify or create data.
- *The administrative employee:* accesses to administrative data. Can record, modify or create new data.
- *The system administrator:* has both rights of the supervisor and rights to set security levels for other users and create or suppress a user.

4.1.2 Use Cases

In this paper, we won't to describe all 'Use Cases' useful of our IS because of the complexity of the system and the number of pages that it may include.

Use Case "system Login/Logout"
- The user enters the system with a 'login' and a 'password' after identification
- **If** the system identifies the user as authorized **Then**
  It allows access
  Otherwise
  It denies access
  It locked at the user identification step
- A user must logout from the system
  - **If** a user is not using the system during a period **Then** The system locks the access
  End if
End if

Use Case "see a record"
- The user selects the option "see" a record
- The user enters criteria to find the record to read
- **If** the record exists **Then**
- The system displays it
- **Otherwise**
- It informs the user
- It waits for another transaction
- **End if**

Use Case "create a new record"
- The user selects the option "new" a record
- The user enters criteria to create the new record
- **If** there is no similar record in the system **Then**
- The record is created by the system
- **Otherwise**
- It informs the user
- It waits for another transaction
- **End if**

Use case "modify an existing record"
- The user selects the option "delete a record"
- **If** the user has the authorization **Then**
- The record may be deleted
- **Otherwise**
- The user is informed
  The system waits for another transaction
  End if

Figure 2 illustrates the functional model of the UFEC School.

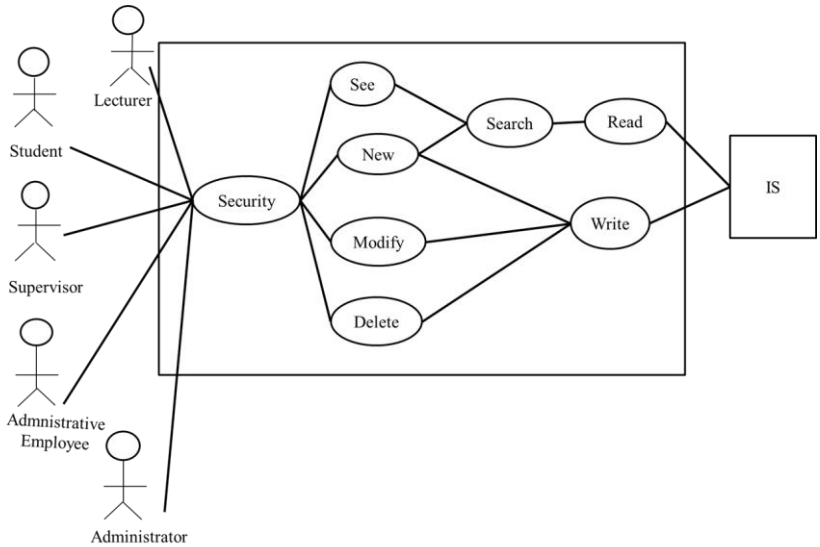

Figure 2: The functional model

4.2 The object model

The Object model (also called the Static model) describes the objects (concepts from the real world) and their relationships: description of the structure and characteristics of the objects (the "Classes"), and description of the various associations between the different objects (the "Associations").

4.2.1 Identification of Classes

Objects are characterized by having both 'state' and 'behavior'. The 'state' of an object is the information an object has about itself. For example, 'a student' may have *a name, a date of birth and a university matriculation number*. The 'behavior' of an object describes the actions the object is prepared to engage in. For example, we might ask 'a student object' for its *age*. This would involve the 'student object' performing a calculation based on *its date of birth* and *today's date*. The behavior of an object is described by the set of operations it is prepared to perform.
A 'graduate school' or a 'university' would typically have a large number of students. Unlike, real students, all 'student objects' exhibit the same behavior and carry the same knowledge about them-selves. We might model a 'student object' with *a name, date of birth* and *matriculation number*. The actual state values for 'two student objects' are different since 'university' or 'graduate school matriculation numbers' are unique. With a large 'university' or 'graduate school' population, we might, however, expect two or more students with the same *name* or two or more with the same *date of birth.*
How can we resolve this problem? All of our 'student objects' support a single abstraction that we may choose to call 'student'. We refer to the abstraction as the Class of the object.
The Class describes any number of 'student objects'. The 'Supervisor Class' describes any number of 'supervisor objects'. The Class describes the information and object holds to represent its state. The items of information are called *attributes (or properties).* The Class also defines the behavior of such objects, listing the operations they can perform, i.e. the messages they can receive. The effect of these operations is described by its *method.*

4.2.2 Identification of relationships

Objects enter into relationships with each other. One of the most important relationships is "Association". In general, "Association" should be used where two objects are not conceptually related but within the context of the problem need to make use of each other's services. For example, an interaction in which a single 'Administrative Employee' object is employed by a single 'Graduate School' object. They are associates in the sense that the 'Graduate School' object adopts the role of the 'Employer' while the 'Employee' object adopts the role of the 'Employee'. The 'Employee' could request the name of the 'Employer' while the 'Employer' could request the job title or salary of the 'Employee'.

4.2.3 The Class diagram

The Class Diagram lies in the fact its content delivers the primary elements in our program code, namely the Java Classes. A Class Diagram describes the types of objects in the system and the relationships that exist between them. A Class Diagram is an abstraction for all the possible object diagrams we might construct. A Class is also documented with its set of attributes and operations. The attributes represent the set of values each instance maintains the object's state. The set of operations are the messages an object of the Class may receive.

Figure 3 illustrates an extract of the Class Diagram of our IS.

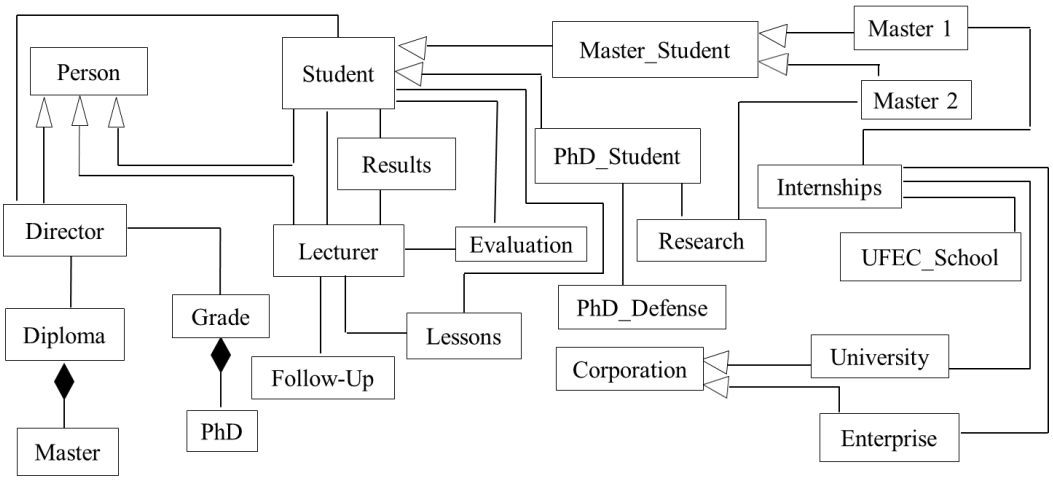

Figure 3: Extract of the 'Class diagram' of the UFEC School "Training Department"

4.3 The dynamical model

The dynamic analysis phase is an important step in the definition of objects and understanding of their functioning. It aims to describe:

- Temporal and event relationships between system objects described in static models.
- Objects state i.e. internal changes during the course of the application depending on the options selected by users.
- Actions performed by objects in a given context.
- External actions of the system on objects in the studied system and the reactions of these objects.

Dynamic modeling is based on several models aimed firstly, to describe the interactions between objects in the system and external systems, and secondly, to study the evolution of internal objects.

We illustrate one type of UML2 diagrams (called sequence diagram or scenario) of our application: the "creation of a new student".

Figure 4 illustrates a part of the dynamical diagram.

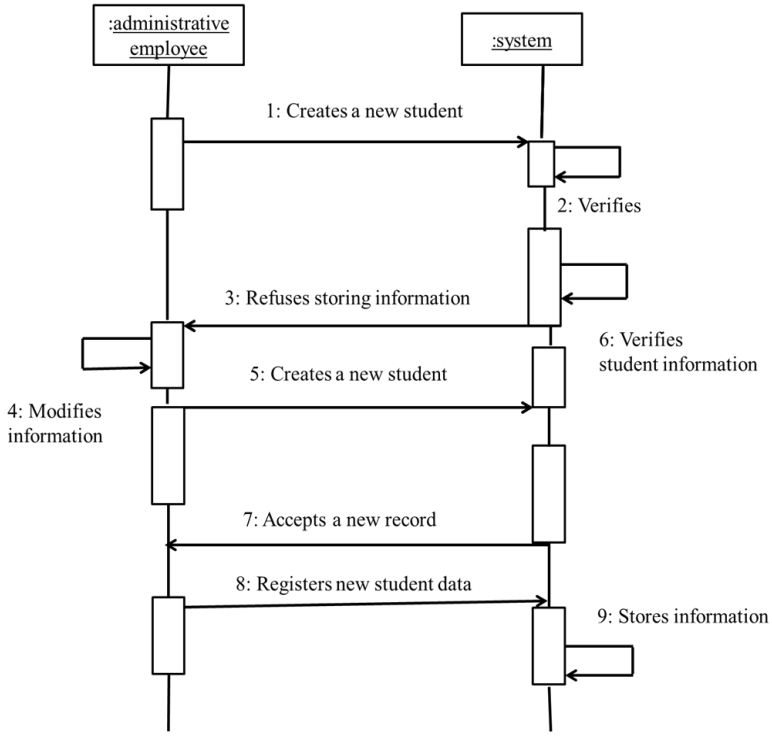

Figure 4: The dynamical diagram ('register a new student')

Section 5 is the conclusion of the different concepts underlined in the paper.

5 Conclusion

In this paper we have developed an Object-Oriented model with the UML formalism in order to implement a real Information System in a graduate school in Paris (France). This work is a part of a type of research conducted in our School. This model has summarized all concepts and knowledge used for the organization of the school. These concepts were formalized with various diagrams according to the Object-Oriented approach with the unified language UML (UML2). The approach permitted us to structure the management of the school which was manual except some electronic files with Excel and Word. The school is a new organization and opened since only four years (2008). In a few years, the school has developed a lot of academic sectors of learning and training. It needs a re-organization and restructuration of knowledge for the management of the school with a better management based on high and modern technologies. The Object-Oriented approach was suitable to manage the complex and various sources of data providing through a multiple of departments. The school has a multimedia department which manages image data. The approach is completely suited. Classes will help us to generate automatically the code and implementation will be easy to complete. We project in a near future to implement the system and place it in the graduate school as soon as possible.

6 References

[1] Mary Culnan. “Mapping the Intellectual Structure of MIS”, 1980-1985: A Co-Citation Analysis”, MIS Quarterly, Vol. No.11, Issue No.3, (341-353), (Sept 1987).

[2] Peter Keen. “MIS Research: Reference Disciplines and a Cumulative Tradition”, in Proceedings of the First International Conference on Information Systems, McLean, E.R. (Ed.), (9-18), (Dec 1980).

[3] Allen Lee. “Architecture as A Reference Discipline for MIS”, in Information Systems Research: Contemporary Approaches and Emergent Traditions, H.-E. Nisen, H. K. Klein, and R. A. Hirschheim (eds.), (573-592), (1991).

[4] John Mingers and Frank Stowell. “Information Systems: An Emerging Discipline”, (eds.). McGraw- Hill, (1997).

[5] Mark Silver, Lynne Markus and Cynthia Mathis Beath. “The Information Technology Interaction Model: Foundation for the MBA Core Course”, MIS Quarterly, Vol. 19, Issue No.3, (361-390), (Sept 1995).

[6] Robert Galliers, Linne Markus and Sue Newell. “Exploring Information Systems Research Approaches”, (Eds), (2006).

[7] Claudio Ciborra. “The Labyrinths of Information: Challenging the Wisdom of Systems”, Oxford University Press, (2002).

[8] Alan Hevner, Salvatore March, Jinsoo Park and Sudha Ram. “Design Science, in Information Systems Research”, MIS Quarterly, Vol. No.28, Issue No.1, (75-105), (2004).

[9] Salvatore March and Gerald Smith. “Design and natural science”, in Information Technology (IT), Decision Support Systems, Vol. No.15, (251-266), (1995).

[10] David Avison and Steve Elliot. “Scoping the Discipline of Information Systems”, in Avison, D. E. and Pries-Heje, J. (eds), (2005).

[11] Mokrane Bouzeghoub, Georges Gardarin and Patrick Valduriez. “Les Objets ”, Eyrolles, (1998).

[12] Laurent Debrauwer and Naouel Karam. “UML2: Entraînez-vous à la modélisation”, Editions ENI, (2006).

[13] Kenneth Barclay and John Savage. “Object-Oriented Design with UML and Java”, Elsevier Butterworth-Heinemann, (2004).