An Object Model of the multipurpose cadastral information - a case study

Moha EL AYACHI, Mohamed ETTARID, El Hassane SEMLALI and Driss Tahiri, Morocco

Key words: Approach, design, multipurpose cadastre, analysis, modeling

SUMMARY

The GIS technology evolves rapidly to encompass different domains of the real world from the simplest applications to manage a series of data for determining some statistic results, to the huge and complex systems serving to analyze and make decisions. Critical efforts have been made to improve the computer engineering goals and the designing approaches to build GIS applications. To every problem are associated specific GIS solutions such as the multipurpose cadastral information system, which needs reliable management to enhance land management, land ownership administration, land resources assessment, network infrastructure studies, and land valuation. The structure of cadastral information with their large database is not easy to manage using conventional methodologies. The need of introducing the object-oriented approach is very significant, because of their ability to integrate textual and geographical features and to deal with their various aspects in a seamless manner. This paper has the aim to describe a method based object oriented approach to design a multipurpose cadastral information system. Two major reasons are considered. Firstly, the GIS technology becomes increasingly sophisticated including new software packages that support the OOA concepts. Secondly, this new approach will be a good beginning to implement an integrated multi-purpose cadastral system based GIS. As a result of this study, one can find that the Object Oriented approach provides powerful tools to express suitable analysis and modeling of the real world. However, in terms of implementing phases, the technique has significant weaknesses. Thus, it is required to involve the Unified Modeling Language to provide best solutions for implementation issues.

An Object Model of the multipurpose cadastral information - a case study

Moha EL-AYACHI, Mohamed ETTARID, El Hassane SEMLALI and Driss Tahiri, Morocco

1. INTRODUTION

The multipurpose cadastral information system serves as a reference framework to enhance land information management and promote the relationship between people and land (McLaughlin, 1995). The relationship enhancement depends largely on the development of a convenient multipurpose cadastral information system as a comprehensive system offering several advantages in managing all kinds of land information (El-ayachi, 2006). The development of a such system is acheived by analysing the real world, designing the new outlook, and implementing the solution. The implentation process uses the GIS technology as a power tool to computerize an existing system in order to operate its various activities such as collecting, analyzing, processing, updating, and retrieving data. Major efforts have been made to support and improve the GIS engineering goals and principles in terms of suitability, accessibility, reusability, and reliability.

Every GIS, before being implemented must be designed to facilitate data analysis and management. Various levels and approaches are developed to deal with this subject. The levels may be grouped into three stages: conceptual, logical, and physical level (Longley et al., 2001). The conceptual level involves the tasks of identifying the real organization and its functions to be computerized. It also permits to define the structure of handled data, the relationships between them, and the user views of the future system. The logical level concerns studying the data types to be used in specific GIS software. It enables structuring geographic data to be easily implemented in a computer. The physical level constitutes the final step aiming to define the physical structure of the GIS to develop. The approaches used to design GIS in general and particularly the land information systems may be divided into two major categories. Firstly, we find methods based on waterfall Life Cycle and secondly the emerging methods based on the large concepts of the Object-Oriented Approach (OOA).

In this paper, a new Object Model for a multipurpose cadastral information system in Morocco is conceived. To illustrate this goal, we describe the requirements and perspectives of the future computerized system. Basic concepts linked to the object oriented approach and to the object modeling technique are discussed to provide a deeper understanding of the OOA and of the new system. The presented results are limited to the analysis phase within which we describe the object, dynamic, and functional modeling. The designing and implementation phases will be developed in a national commitment in accordance with various competencies and experts' visions in Morocco. The conceptual solution achieved in this study will be a key or further research and a good way to inspire designing and implementation enhancement.

2. THE WATERFALL LIFECYCLE VERSUS OBJECT ORIENTED APPROACH

The waterfall Life Cycle design assumes that the development of an information system flows through a series of stages such as: analysis, design, implementation, testing, and maintenance. The stages are mostly sequential because each stage in the "waterfall" model must be requisitioned while the preceding stage is complete (Figure 1). Each stage has its own method to be released. For that reason there are barriers from stage to stage (Taylor, 1992).

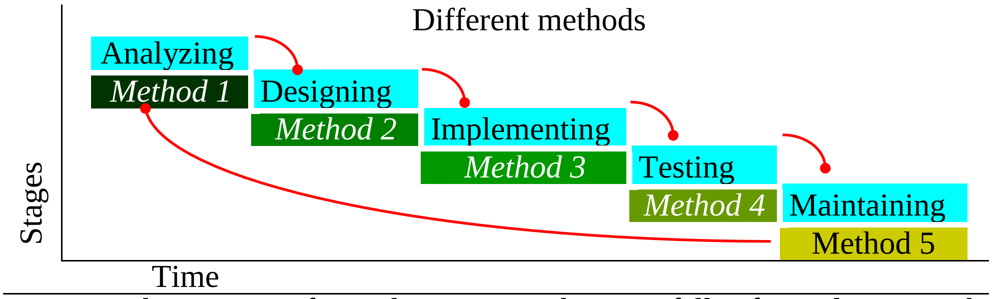

Figure 1: The concept of GIS design using the waterfall Life Cycle approach

The Object Oriented Approach enhances the Life Cycle design in two manners by using three basic mechanisms, which are: objects, messages, and classes (Haining et al., 1995). It reduces the barriers between stages using a unique language and provides specific tools to assist each stage. The real world objects are described in the object analysis phase and translated directly into system objects in the design object phase, and so on (Figure 2). This way makes a system easier to be tested and maintained (Muller, 1999).

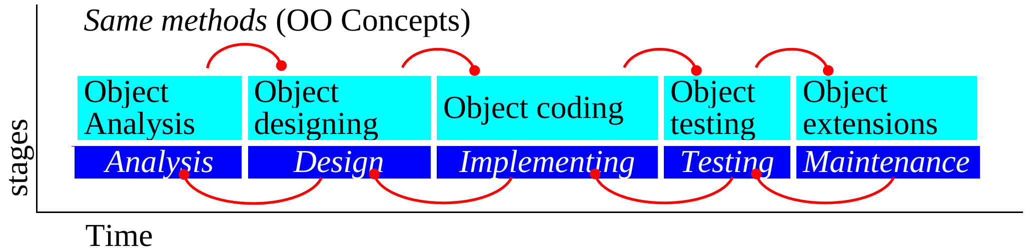

Figure 2: The concept of a GIS design using the Object Oriented Approach

The structure of cadastral information with their large database is not easy to manage using the first methodologies. The involvement of the OOA will enhance the establishment process because recent GIS technology becomes increasingly sophisticated by including new software packages that support the OOA concepts. It integrates textual and geographic features and deals with their different aspects in a seamless manner. Several studies have used this approach to develop power applications in the domain of cadastral information system and gaining good and practical results (Dueker, 1986; Dueker, 1987; Dueker, 1990; Tuomaala and Uimonen, 1998; Li et al., 1999).

The OOA is supported by various Computer-Aided Software Engineering (CASE) tools, which generate easily and directly the object physical model from the logical model. Once a CASE tool creates and specifies the physical model, the intended object application will be put into operation. Among existing object-oriented methodologies, the Object Modeling Technique for (OMT) comes up as one of the most popular that permits understanding the real world by analyzing its static and dynamic components and designing a new system behavior (Rumbaugh et al., 1991). This technique has been involved in many real world problems because of its simple core notation and continues recently to be successful in several domains such as telecommunication, road network, and urban studies.

3. THE OBJECT ORIENTED APPROACH PRINCIPLES

3.1 The general concepts

The term “Object Oriented” means the organization of a software system as a collection of discrete entities that integrate both data structure and behavior. The word “oriented”, defined as directed toward, has the aim to caste the "object oriented" into an adjective meaning object directed toward any thing we can think of (Page-Jones, 2000; Eckel, 2000).

An object is something that has a definite set of characteristics, a set of things it does, and a current state. It encapsulates related data and functions called methods, into unique package. Each object is accessed only through its methods. If we deal with an owner, its characteristics are captured in what we denote variables (owner identifier, owner name, owner address, owner account, etc.). The operations that owner does are formulated in its methods (report name, update address, report account, etc.). The state of an object indicates something that is internal to the object.

To establish interaction between objects, the concept of message is introduced as a simple request from one sender to another receiver. The message allows objects to interact without having to interfere with each other's internal processes. The encapsulation of data and methods within an object makes the object much easier to be modified. The internal structure of an object may be changed the manner its methods work, but if the messages it responds to don't change, nothing else in the system is affected by these changes (Dale, 2000). Another benefit of messages is the use of polymorphism principle, which means using the same name for different operations in different circumstances.

Objects with similar characteristics are grouped into a specific type of objects called classes, which are the basic unit of the analysis in the OOA. The general characteristics of similar objects in one class have to be determined rather than in every object. An individual object of a class is called an instance of the class. Many objects from the same class may be created to constitute instances of that class (Ralston, 1995).

The class relationship is described in three ways (Figure 3): association, aggregation, and inheritance. The association defining the conceptual connection between classes, is depicted by a diagram that often includes the verb describing the relationship between them. The aggregation indicates a relationship meaning "is composed of". It is a specific form of association but it strongly enables to represent the connection between classes through bi-directional ways. The inheritance usually indicating a relationship like "is a kind of" is a mechanism showing that the special cases known as subclasses share all the characteristics of their general cases known as super classes. The subclass inherits many properties of a super class. The property that is unique to a particular subclass is defined in the class itself.

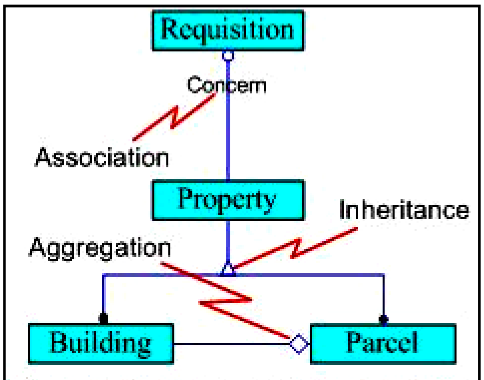

Figure 3: The class relationship notation

3.2 The Object Modeling Technique concepts

The Object Modeling Technique (OMT) is among the object-oriented methods to construct a series of models. It avoids the rapid approach of prototyping applications allowing to develop small software, gradually evaluated and made robust through incremental enhancements, but with less efficiency. Currently, numerous books describing the OMT exist, but it is useful to refer to the original significant book written by James Rumbaugh et al. (1991) that provides a practical and productive way to develop object oriented-based applications. The OMT process can be divided into analysis, design, and implementation phases.

The analysis phase enables to understand the requirements and the real world environment to computerize. It constitutes the first step of the OMT methodology serving as a framework for design and implementation phases. The analysis provides the designer with precise and concise model to abstract the most important features of a real world. The main steps of this phase are object, dynamic, and the functional models. It begins with the problem statement, which specifies what is to be done and how. It is concerned by collecting needs generated by clients, managers, and/or developers. This means to establish a series of requirements statement including: The problem scope, the needs, the application context, the assumptions, and the performance needs. The analyst avoids the design and implementation considerations for they restrict flexibility.

The design phase is built upon the produced models during the analysis phase serving as the skeleton. It is based on the analysis of the existing structure and intends to realize a new architecture. It allows breaking the system down into subsystems and selecting appropriate algorithms to be used (Hunt, 2000). There are two processes in this level: the system design, which determines the overall system architecture and the object design refining and optimizing the analysis model from applications to computer concepts.

The implementation phase consists of coding the system and object designs into the target software. It provides very useful information on how to develop the desired application. This stage is composed of the coding process, which implement classes in the target software and the testing process, which consists of testing each phase of all the process.

4. A CONCEPTUAL MODELING OF THE MOROCCAN CADASTRAL SYSTEM

4.1 The ongoing efforts in Land Administration modeling

In 1998, the Cadastre 2014 booklet was published to describe a new vision for the future cadastral system based on six statements (Kaufmann and Steudler, 1998). The authors concluded in the statement four that the modern cadastral systems have to provide the basic data models and invite surveyors to apply modern technology for handling the models. Regarding the importance of the booklet, a first Moroccan Arabic version was established to provide government and stakeholders with new trends in land administration (El Ayachi, 2001). In 2002, a study was conducted in Morocco to design a computerized cadastral information system. An initial model based on the entity-relationship model was developed. However, the model had known several shortcomings (El Ayachi, 2006). We distinguish the difficulties linked to the use of various platforms managing cadastral data, the slowness of the update, recovery and storing processes of data in the existing system, and the dispersion of land information between textual database and graphical storage files of cadastral entities.

On the other hand, the cadastral map, which is a dynamic basic component for the cadastre reflecting the daily changes affecting lands, allows visualizing only information linked to the titled properties in a geographical area without enabling the decision makers to carry out spatial analyses. To overcome the previous challenges, it was recommended to set up a cadastral system that permits to unify the source of information and to process any kind of spatial information related to the cadastral data. The problem was how to define the suitable structure of a future cadastre as a referential encompassing all the development projects.

Since 2006, various efforts have been made to investigate the state of the art of the land administration domain model developments. Basic international models are examined (Oosteron Van and Lemmen, 2002; Lemmen and Oosteron Van, 2006; Van Bennekom-Minnema, 2008, Hespanha et al., 2008) to take benefits of the worldwide modeling strategies and the results were very appreciated and a study was initiated recommending the use of the object oriented approach to design such system.

4.2 The problem statement

In the analysis phase, it is required to collect the maximum of information related to the description of the existing systems. The description consists of defining the semantic aspects of the multipurpose cadastral information system. To achieve this goal, a series of meetings and visits are organized within the cadastral services of six major cities in Morocco. The technicians, engineers, and administrators of these services are interviewed for enhancing the establishment of the problem statement. A series of undertaken projects are analyzed. We remarked the use of various platforms, within the Moroccan cadastral services, to develop pilot projects such as AutoCAD Map, Map/Info, ArcView GIS to develop prototypes for cadastral purposes. Other software packages are tested namely Microstation with Oracle. Their objectives behind these projects were to establish computerized applications enabling management of cadastral attributes concerning each cadastral task within a service of cadastre. They also aimed to manipulate geometrical and textual data in the same frame.

It is concluded that there is a major need in developing a multipurpose cadastral information system (MCIS) dealing with land information, natural and physical resources data, and planning information in both their geometrical and textual forms. The use of the Object Modeling Technique, to develop such system, is for a great importance. The useful basis to implement such system is the conceptual process.

In Morocco, the cadastral system is interacting with various users indicated as customers. They can be classified into four categories, government institutions, private agencies, individuals, and scientists. The government institutions encompass major public agencies and state administration generally using cadastral and cartographic products. We distinguish the directorate belonging to the ministry of housing and territorial management, ministry of agriculture and marine fishing, ministry of equipments and transportation, and ministry of Islamic affairs, etc. The public agencies cover several offices linked to water management, power management, mining administration, airports, and railways, etc. Private agencies involved in land and natural resource management are willing to obtain up-to-date cadastral information and maps to integrate them in their project development and management. Individuals, who could be owners or none, are claiming various needs related to the cadastral information to process in a right and rapid way their request such as land ownership transferring, land market transaction, and land right maintaining, etc.

Those are considered potential customers of the cadastral system. They claim a series of needs, which include the following elements:

- Obtain information on properties, parcels, and ownerships,
- Show geometrical data and attributes on properties and parcels in a user-friendly interface,
- Select a zone to show its natural, physical, and cadastral characteristics,
- Combination of attributes and graphical data in the same frame,
- Consult of historical information concerning properties, parcels, and ownerships,
- View urban information and all related services,
- View recreational areas, environmental sites, market places, and various parks within a city, town, or neighborhood,
- Get location by introducing site name, road intersection, or complete address, or eventually coordinates,

- Get textual information by clicking on screen a graphical object such as linear, zonal, and punctual details,
- Show district, provincial, and regional administrative boundaries,
- Locate geodetic details concerning monumented points and precise leveling benchmarks,
- Get several maps concerning cities, towns, and/or rural areas?

The implementation of a such is based on a bunch of requirements. The system should integrate both GIS software and a database management system (DBMS). The GIS tool would include various products enabling the management of geographical features associated to their attributes. The DBMS would permit to store and manage attributes related to cadastral, natural, and physical resources. The system should provide managers and users with basic tools for querying, editing information, sharing information, and updating data.

4.3 Building the object model

The object model constitutes the first step in analyzing the real world. We performed several steps namely, identifying objects, classes, and associations, constructing data dictionary, refining the model by simplifying classes and using inheritance, and structuring classes into modules. The necessary classes extracted from the real world and classes identified regarding the customer claims are summarized with their attributes and operations in the object model (**Figure 4**).

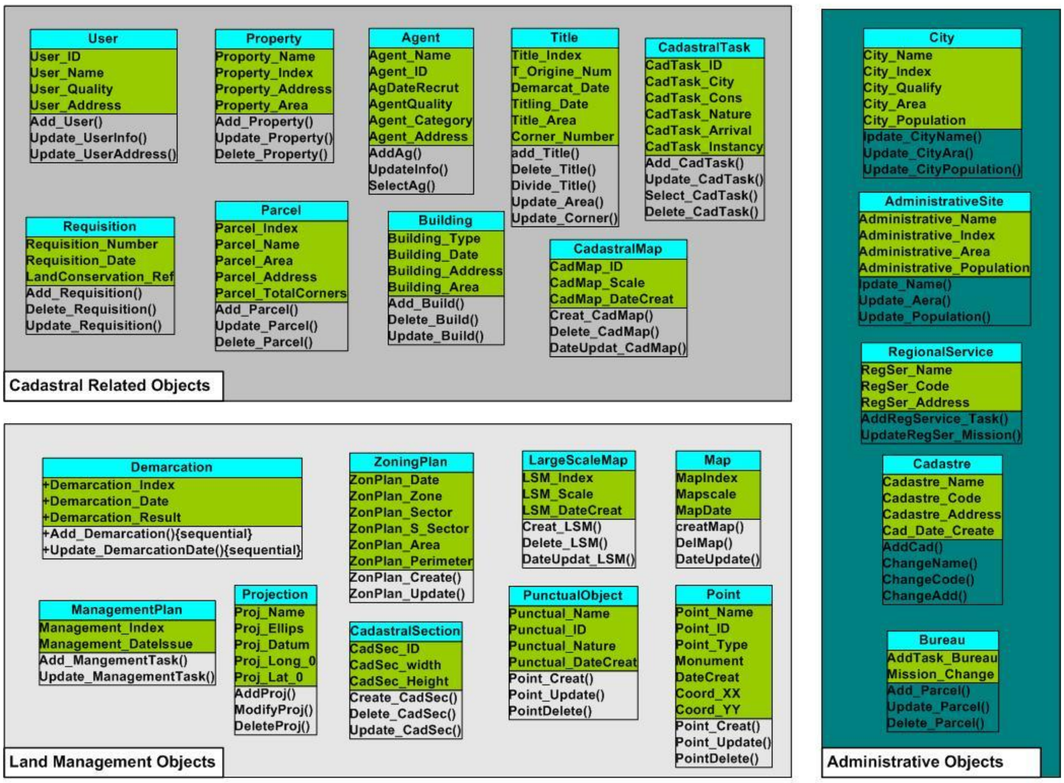

Figure 4: Classes with attributes and operations

A data dictionary, which has the aim to describe each object class, is elaborated to facilitate the understanding of the objects. The dependencies between two or more classes, which are indicated as association are described and identified in a part of the initial diagram (**Figure 5**).

This diagram has been refined to permits building a definitive object model. Attributes are data values held by objects in a specific class. They define the state of an object. Each attribute has a unique name within a class. The inheritance relationship permits the subclasses to share attributes of their super classes. An attribute can have a value that depends on a particular context.

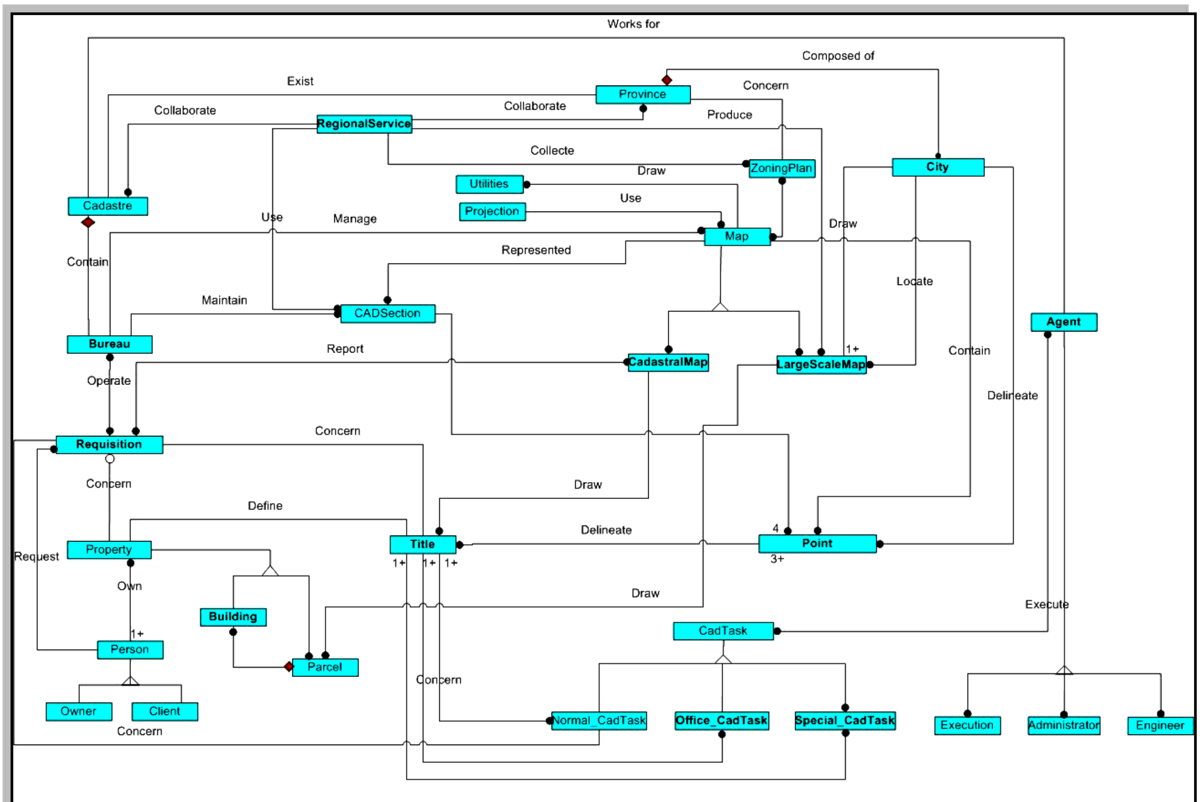

Figure 5: Part of the Initial object diagram of the multipurpose cadastral information system

This value is called qualifier that qualifies an association between two classes. Each title concerning a property is related to a cadastral map by its index called "TitleIndex". This qualifies the association between "Title" and "CadastralMap". The same analysis is fully right for qualifiers "RequisitionNumber", ", "AgentID", and "CadSecID" qualifying respectively the classes "Requisition", "Parcel", "Agent", and "CadastralSection" (**Figure 15**). Some classes are redundant such as "client" and "owner" because we cannot find attributes that characterize those classes. The same remark is convenient for the subclasses of "Agent". The **Figure 15** presents the object model for the designed system after a series of refining.

4.4 Developing the dynamic model

The dynamic model shows the time dependent behavior of the system and the objects in it. This model is insignificant for a purely static data management such as a database. It is important for interactive systems. The MCIS has an interactive interface dominated by the dynamic model, which consists of four steps that are: preparing scenarios for interaction sequences, identifying events between objects, preparing event trace for scenarios, and building a state diagram.

4.4.1 Preparing scenarios

The first step concerns the construction of normal and abnormal scenarios of interactions between the system and entities outside the system boundaries. A normal scenario is just a sequence of events that causes an object to change state. An event occurs at any time information comes into the system or goes out from the system. An abnormal scenario also called a scenario with exception includes error conditions, incorrect inputs by users, omitted inputs, data values outside the range of valid values, and where the time out will occurs. A set of scenarios for the MCIS has been identified based on the discussions with users. The scenarios are: Authorize access to the system, Update data, Searching for, or Finding information, Consult data, and inquiries by Request process.

Scenario 1: Normal scenario for authorizing access to the system

- A user sends a request to use the system.
- The system asks for authentication.
- The user introduces his name and password.
- The system sends information to the Administrator.
- The Administrator assigns an access authorization.
- The user proceeds to using the system.

Scenario 2: Normal scenario for updating data

This scenario concerns only internal agents that have the right to add, modify, or delete data of the MCIS.

- The agent introduces his name and code to access to the system.
- The system controls the authentication and authorization, if the agent is accepted.
- The agent selects an object or many objects to manipulate
- The agent selects the kind of operation (add, modify, delete),
- The agent operates the action and verifies the new data,
- The agent selects other objects or cancels to end the session.

Scenario 3: Normal scenario for searching or finding information

This scenario concerns an external user that needs to search for or finds multipurpose cadastral information that can be a location by its coordinates or address.

- The user selects the search option to start the request.
- The user enters a convenient word, phrase, one or more keywords describing better the location.
- The user requests the system to start search.
- The system returns the information satisfying the search criteria by locating graphic information on the screen and/or visualizing a set of information.
- The user views the information
- The user processes another search operation or ends the session.

Scenario 4: Normal scenario for consulting data

This scenario has the aim to enable a user to consult cadastral, urban, juridical, or environmental data.

- The user selects the consult option to start the request.
- The user selects kinds of data to consult (cadastral, urban, juridical, or environmental).
- The system asks the user to enter his name and password.
- The user enters the name and the password.
- The system verifies the user’s information, accepts, or rejects the user.
- In case of acceptance, the user enters the exact word, index, name, or reference of the requested data.
- The user requests the system to start the consultation.
- The system returns the results of the operation satisfying the criteria by visualizing graphic data on the screen or/and a set of data.

- The system asks the user to select the kind of data delivery (file, map, or a document). The document can be a certificate or a regulation description of urban zoning or environmental matters.
- The user selects the option delivery or cancels the process.
- The system processes the action.
- The user ends the request for consulting data.

Scenario 5: Normal scenario for request process

This scenario has the aim to enable a user to request the system and process various operations according to specific criteria.

- The user selects one or more features to visualize.
- The user starts the request.
- The system returns the view containing the requested features.
- The user makes another request or ends the process.

4.4.2 <u>Events, event traces, and flow diagram</u>

The internal computation steps are not events, except for decision points that interact with the external devices. Events include signals, inputs, transactions, and actions to or from users or external devices. For each event, we identify actors that cause this event. Each type of event must be allocated to the object classes that send it and receive it. An ordered list of events between objects is expressed by an event trace that concerns each scenario.

Each object is assigned to a column in a table. The **Figure 6a** represents an event trace of the normal scenario authorizing the access to the system MCIS. The event trace of the scenario updating data is represented by the **Figure 6b**. In this event trace, the interface is considered as a mean to communicate between an agent and the central system MCIS. All event traces are established for various scenarios such as for the scenario of searching or finding information regarding some criteria, the scenario of consulting data.

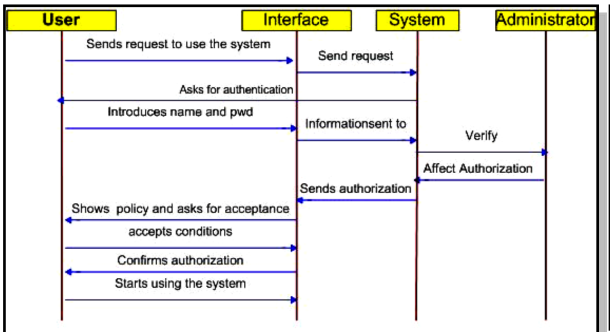

Figure 6a: Event trace for authorizing access to MCIS

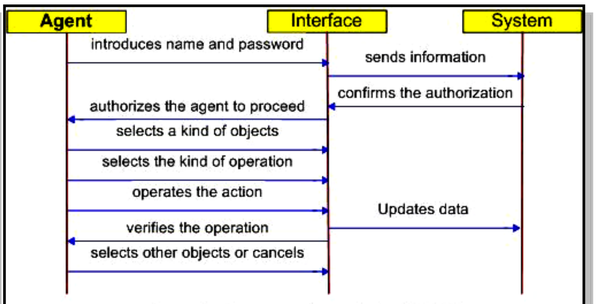

Figure 6b: Event trace for updating MCIS data

Regarding the nature of action each scenario processes, the scenario searching information consists only of locating information that responds to some criteria such as searching a location using its name, address, or coordinates, but the scenario consulting data is reserved for authorized users to obtain data concerning their properties, services, urban information regulating their living area, or requesting some documents such as certificates and maps. The event trace lied to the scenario request process presents the possibility of using system to perform some visualization operations such selecting specific features, zooming, or overlaying different features and layers.

For summarizing events between classes, we use the event flow diagram without regard for sequence. This diagram constitutes an overview for all the event traces in the system. Its usefulness is linked to the fact that it gives a software engineer a high level of the possible control flows in a system.

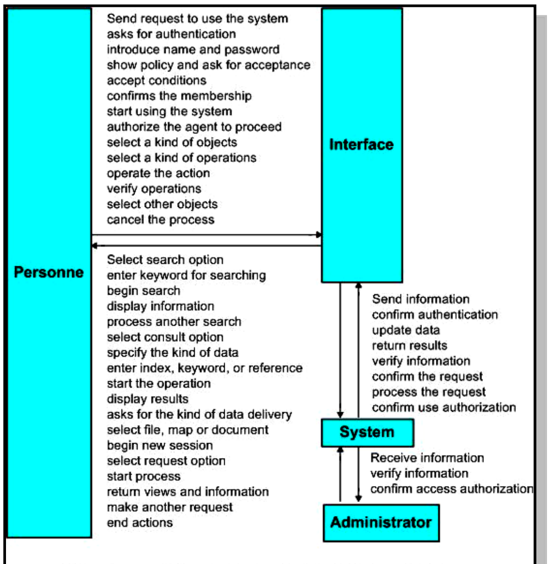

Figure 7: Event flow diagram for a multipurpose cadastre

The **Figure 7** shows an event flow diagram for the MCIS. It includes events from all scenarios, including errors events.

4.4.3 Building a state diagram

The state diagram will be developed for object classes that have nontrivial dynamic behavior. Through the state diagram, we find that every event trace or scenario corresponds to a path. The diagram should cover all scenarios and handle all events that can affect an object of a class in each of its state. A change of state caused by an event corresponds to a transition. The state is shown as a rounded box containing an optional name. The relationship between users and the system is the user interface. Then an object class called user interface is added to the object diagram. The **Figure 8** shows the state diagram of the user interface.

The object class called 'agent' has its state diagram represented by the **Figure 9a**. The diagram shows the states with rounded boxes and the transitions between states with rows. It describes actions that an agent within a cadastre can make to update data.

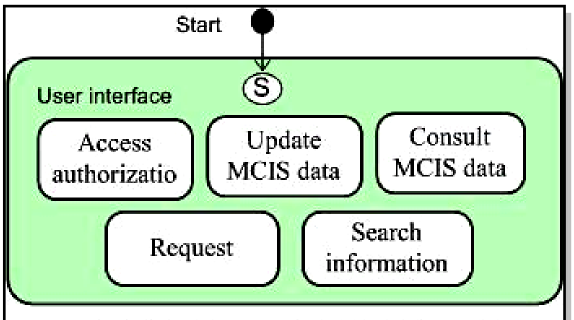

Figure 8: User interface state diagram

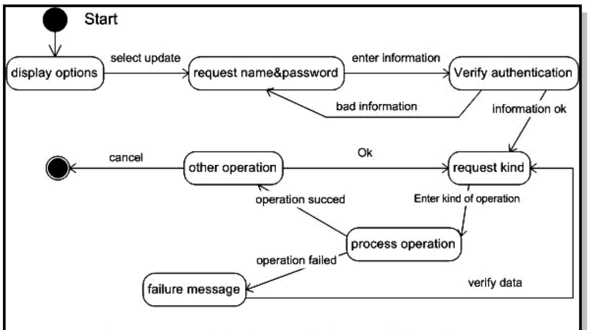

Figure 9a: The state diagram for the object class Agent

This process can encompass several operations such as adding data to the database, deleting data, modifying data, or creating new data from the existing information. Only an authorized agent has to use the update option. This allows increasing the security of the MCIS data.

The state diagram for the object class "user" is represented by the **Figure 9b**. The actor "user" of the MCIS processes three major kinds of operations. Firstly, he can search for data concerning properties, urban features, and natural resources aspects. He can also send request to the system to get interactive information such as selecting data layers or visualizing various features on the screen. Secondly, every user should have an authorization to execute consulting operation or download specific information about cadastral, urban, environmental objects. Thirdly, the user needs an authorization to access to the system. This allows him to use the system for his desires.

After making up the dynamic model, we should update the object model to incorporate the user's interface object class. Based on the scenarios and event traces developed so far, we remark that the user interface class has associations with "agent" class and "user" class. ***So the object model should be revised with the controller object, which is the user interface***.

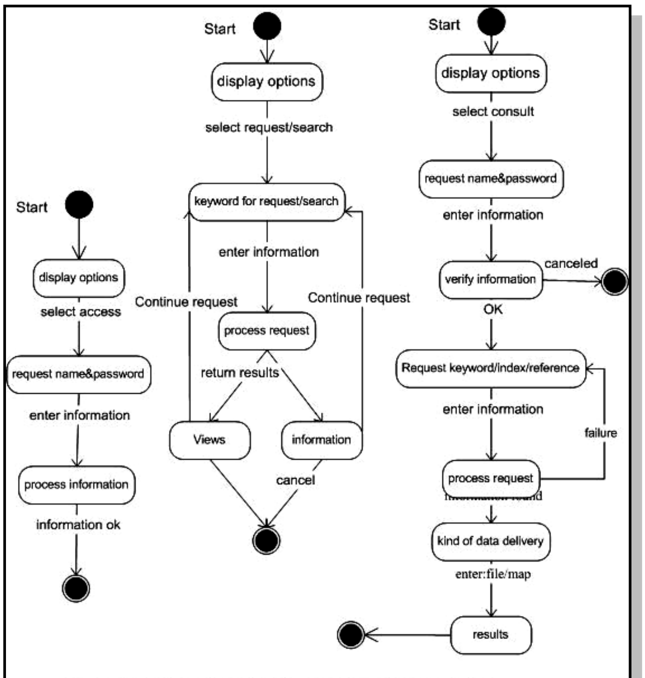

Figure 9b: The state diagram for the object class user

4.5 Constructing the functional model

The functional model constitutes the third step in the analysis phase. It consists of multiple Data Flow Diagrams (DFD) showing flows of values from external inputs, through operations and internal data stores, to external output. The functional model does not inform on how or when the results of computation are happened. It is a network representation of the system showing the functional relationships of the values computed by a system. The data flow diagram is the core unit for this model, which includes for its construction the identification of input and output values, the built of data flow diagrams for each input to output transformation, and the description of functions.

4.5.1 Identifying input and output values

The input and output values are parameters of events between the system and the outside. In our case, all interactions between the system and the outside world pass through the interface. The input and output values are parameters of User Interface events. The **Figure 10** illustrates the input and output values identified for the MCIS.

4.5.2 Building data flow diagrams

A data flow diagram is generally constructed in layers. It is developed by working from outputs to inputs, inputs to outputs, or from the middle of a process towards both the inputs and outputs. In this case, we focus on what a process does and not how to implement it. If a process is doing many things, we should decompose it further until the process performs as a

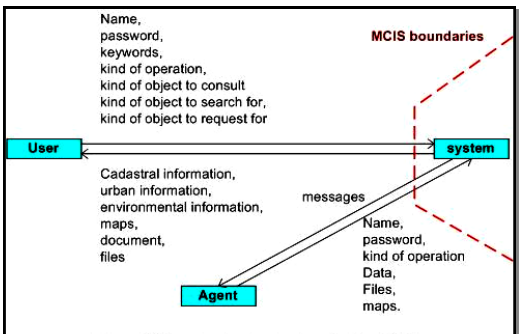

Figure 10: Input and output values for the MCIS

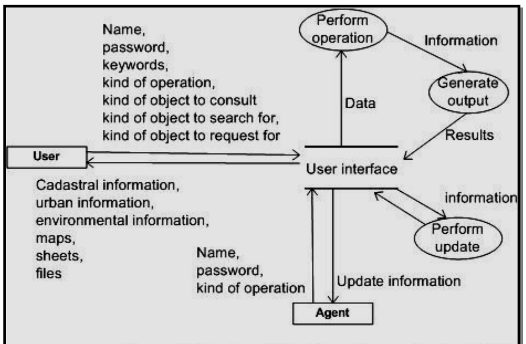

Figure 11: Top-level Data Flow diagram for the MCIS

single function. The **Figure 11** shows the top-level data flow diagram for the MCIS. It shows a data store that is the user interface, in the top-level data flow diagram. The next step is to decompose the processes “perform operation”, “generate output”, and “perform update”. In these processes, each nontrivial process should be further decomposed and expanded.

The **Figure 12** represents a Data Flow Diagram that consists of performing operations selected by a user from the user interface. We distinguish two kinds of important operations. The first one, needing a name and password of a user, enables to consult any kind of data or to access to the system. The second category permits to any user to use the system for finding information or requesting the system. The process “generate output” in the top-level DFD should also be decomposed to enhance the understanding of the application. This process has the aim to enable a user to select a type of output to generate. After processing operations specified by a user, the results may be generated in forms of maps, documents, or screens showing data layers or results of request actions. The process is presented in the **Figure 13**. The process “perform update” shown in the top-level DFD will be expanded to further clarify actions that will affect the cadastral, urban, or environmental data. The **Figure 14** presents data flow diagrams for other processes within the “perform update” process. Generally, processes in the functional model correspond to operations in the object model. They show objects related by functions.

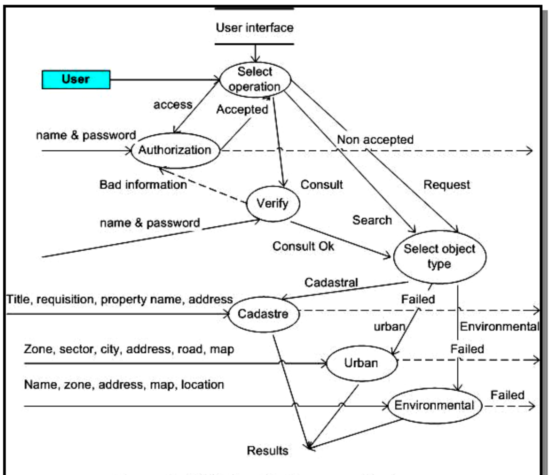

Figure 12: DFD for “Perform operation” process

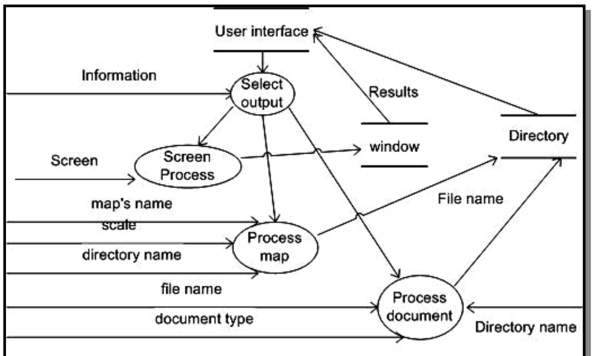

Figure 13: DFD for "Generate output" process

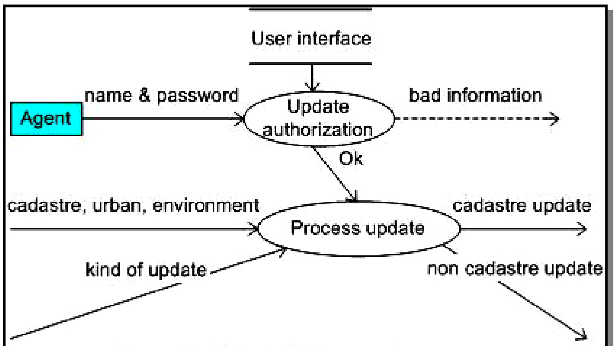

Figure 14: DFD for "Perform update" process

4.5.3 Describing functions

Each process in a data flow diagram corresponds to a function that should be described in natural language, mathematical equations, pseudocode, or decision tables. The description may specify only the relationship between input and output values. This description is called declarative. It may be also procedural if it specifies a function by giving an algorithm to compute it.

4.6 Refining the analysis model

The overall analysis phase may show inconsistencies. We should iterate different stages to refine the three models to produce a design that is more coherent and consistent. New classes and operations will be added to the object model. Various iterations are performed to review

all three models and check for completeness and consistency. New classes are added to object model such as “file” and “directory”. All cadastral, urban, and natural features are linked to these classes. They serve as data stores for data, results, and temporary computations.
From the functional model, by examining the top-level DFD (**Figure 11**), we observe that the processes "perform operation", "generate output", and "perform update" are the key operations. They encompass actions such as consulting, adding, deleting, and modifying. An operation is written in a specific class by preceding the class name or the attribute name by the appropriate action. For example, the operation “add” in the class “agent” is defined as “AddAgent”, the operation “modify a map code” is defined as “ModifyCodeMap”. Other operations can be identified from the object model. They concern reading and writing attributes and association links. These operations are trivial and do not add much value to users. Rumbaugh et al. (1991) suggests leaving these operations out of the object model.

From the dynamic model, the event flow diagram of the MCIS (**Figure 7**) shows events between objects that can be represented as operations on object model. The identified operations are «display» information, «search» one or more object, and "edit" results. These operations may be omitted in the object model and be developed in the object design as functions to implement.

In the previous parts of the paper, we just describe some components of the MCIS and present the steps to follow so as to conceptualize the new system. However, the study was concerned by establishing an object model of the whole system. The **Figure 15** summarizes the refined and final object model for the MCIS. By examining each object and its attributes in the object model, some classes are also omitted such as subclasses of “CadastralTask” and “user”.

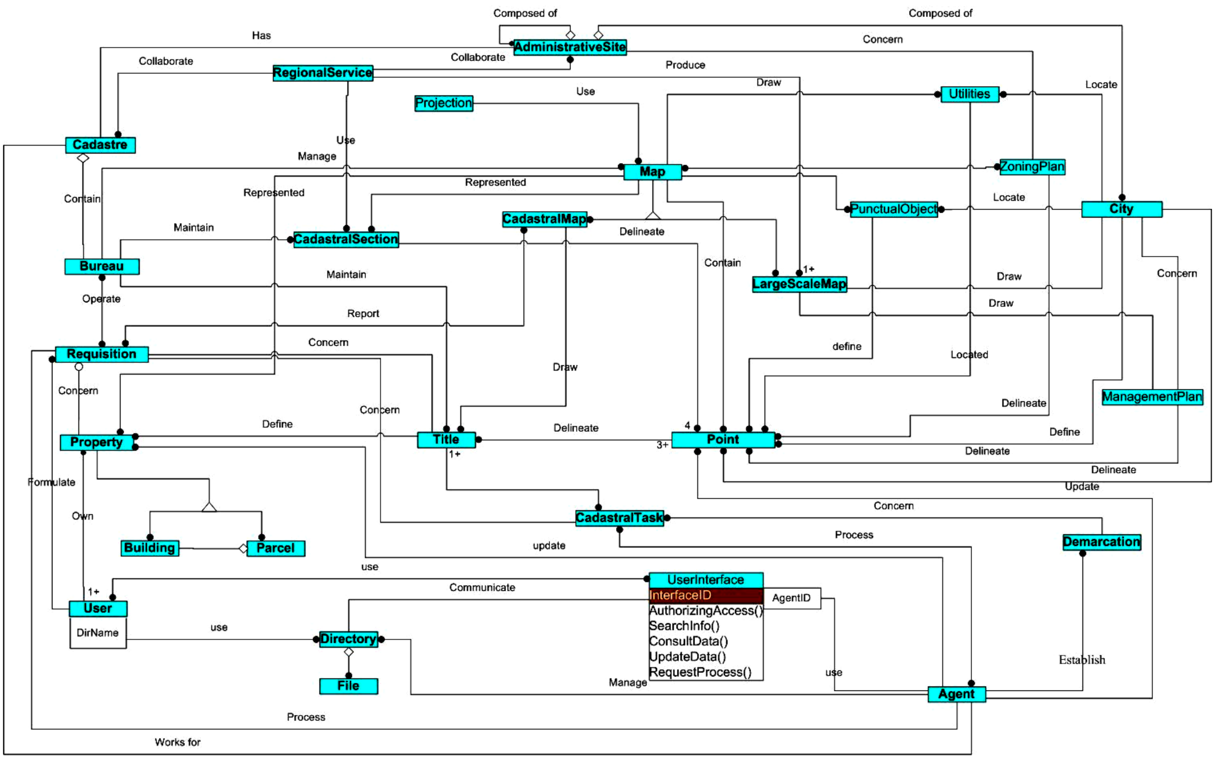

Figure 15: Revised object Model for the multipurpose cadastral information system

The data dictionary is reviewed to show the significance of new attributes and associations. New classes are added and described as below:

- *UserInterface*: constitutes the interface interacting with users and multipurpose cadastral information system,
- *Directory*: a super class reserved for creating and updating files,
- *File*: for storing and retrieving data,
- *PunctualObject:* represents all kinds of punctual features.
- *Utilities:* represents all kind of utilities serving the population such as hospitals, schools, mosques, churches, administration, ..etc.

5. CONCLUSION

The object modeling technique is one of various ways of thinking about problems using models organized around real world concepts. It is based on graphical notations for expressing three main models, which are object, dynamic, and functional models. For a variety of applications, we can use OMT concepts to develop object and oriented tools. The development methodology of OMT consists of building models through three phases, analysis, design, and implementation.

During the study of the real world of the cadastral system in Morocco, we find that OMT provides powerful tools to express a suitable analysis. However, in terms of design and implementation phases, the technique has significant weaknesses. In analysis phase, steps are well defined and structured, while it is necessary to involve a supplement method to conduct design and implementation phases. In the context of the case study, there are two logical ways to develop a multipurpose cadastral information system. Firstly, one may conceive the whole system and describe its static structure using OMT. The used notations during object, dynamic, and functional models allow undertaking an integrated analysis that enhances the reusability of the system. Secondly, the Unified Modeling Language UML can be involved to provide best solutions for implementation issues. Its notations have their origin from three essential methods written successively by Grady Booch, James Rumbaugh, and Ivard Jacobson. Based upon concepts of these methods, a designer can easily and smoothly uses UML concepts to deal with design and implementation tasks. The main advantage of using UML is the availability of software supports to make diagrams and develop source codes in a specified object oriented language. UML is becoming easier by enabling the use of its leading Open Source AgroUML for application design and development (http://argouml.tigris.org/).

REFERENCES

Dale, C.J., 2000, an introduction to DataBase Systems, Addison-Wesley Longman, Inc., 7th Edition, USA.

Dueker, K., 1986, Modeling cadastral spatial relationships using an object-oriented language, 2nd International Symposium on Spatial Data handling, pages 142-157, Williamsvile, NY., IGU Commission on geographical Data Sensing and Processing.

Dueker, K., 1987, A rule-based approach to cadastral mapping using object-oriented GIS, Center for Urban studies, Portland State University, Portland, OR. USA.

Dueker, K., 1990, Modeling cadastral spatial relationships using Smaltalk-80, Journal of the Urban and Regional Information Systems Association, USA.

Eckel, B., 2000, Thinking in C++. 2nd Edition, Vol. 1, USA.

El Ayachi, M., 2006, Design and development of a multipurpose cadastral information system for Morocco, organizational and technical aspects, Doctorat Es Sciences Dissertation, Institute of agronomy and Veterinary Medicine, Rabat, Morocco.

El Ayachi, M., 2001: Cadastre 2014 (Arabic), http://www.fig.net/commission7/reports/cad2014/publ.htm#ara.

Haining, R., 1995, Designing spatial data analysis modules for geographical information systems, Spatial Analysis and GIS, S., Fotheringham and P., Rogerson (Eds), Taylor&Francis Publichers. Great Britain.

Hespanha, J., Van Bennekom_minnema, J., Oosteron, Van P., and Lemmen Chrit., 2008, The model Driven Architecture Approach Applied to the Land Administration Domain Model Version 1.1- with focus on constraints specified in the Object Constraint Language, Integrating Generations, FIG Working Week 2008, Stokholm, Sweeden.

Hunt, J., 2000, The unified process for practitioners, object oriented design, UML and Java, Practitioner Series, Edition Springer, London, UK.

Kaufmann, J. and Steudler, D., 1998, Cadastre 2014 - A vision for a future cadastral system, FIG, 1998.

Lemmen, Chrit., and Oosteron, Van P., 2006, Version 1.0 of the FIG Core Cadastral Domain Model, Shaping the Change, XXIII FIG Congress, Munich, Germany.

Li, J.; Jing, N.; Sun, M., 1999, An object-oriented data model integrating field and object, Geoinformatics and socioinformatics, the proceeding of Geoinformatics'90 conference, The Association of Chinese Professionals in GIS, Ann Arbor.

Longley, Paul A.; Goodchild, Michael F.; Maguire, D.J.; Rhind, D.W., 2001, Geographic Information systems and science, 2nd printing, March 2002, John Wiley & Sons, Ltd. NY, USA.

McLaughlin, J.D., 1975, The nature, function, and design concepts of multipurpose cadastres, PhD dissertation, University of Wisconsin, Madison, USA.

Muller, Pierre A (1999). Modelisation Objet avec UML. Editions Eyrolles. France.

Oosteron, Van P., and Lemmen, Chrit., 2002, Towards a standard for the Cadastral Domain: Proposal to establish a Core Cadastral Data Model, COST Workshop, Towards a Cadastral Data Model, Delft, the Netherland. http://www.14.auc.dk/costg9/.

Page-Jones, M (2000). Fundamentals of object-oriented design in UML. Object technology series, Dorset House Publishing. Edition Addison-Wesley. USA.

Ralston, B.A., 1995, Object Oriented Spatial Analysis, Spatial Analysis and GIS, S., Fotheringham and P., Rogerson (Eds), Taylor&Francis Publichers. Great Britain.

Rumbaugh, J.; Blaha, M.; Permerlani, W.; Eddy, F.; Lornsen, W., 1991, Object Oriented Modeling and Design, Prentice-Hall International Editions, NJ. USA.

Taylor, D.A., 1992, Object Oriented Information systems, planning and implementation, John Wiley &Sons, Inc. USA.

Tuomaala, J., and Uimonen, M., 1998, Introducing the new Object-Oriented Cadastral Information System (JAKO) of Finland, FIG XXI Congress, commission 7, cadastre and land management, Brighton, England. July 19-25, 1998.

Van Bennekom-minnema, J., 2008, The Land Administration Domain Model 'Survey Package' and Model Driven Architecture, Master of Science Thesis, OTB Research Institute, Delft University of Technology, The Netherland.

BIOGRAPHICAL NOTES

Moha El-ayachi, Ettarid Mohamed, Semlali El Hassan, and Tahiri Driss are Engineers of Surveying and Professors within the School of Geomatics and Surveying Engineering at the Institute of Agronomy and Veterinary Medicine. They are active members of the corps of Licensed Surveying Engineers in Morocco (ONIGT). They published many papers in the field of land information, land governance, geomatics, global positioning system, GIS development, and capacity building. They are involved in many partnership searching programs at the national and international levels.

CONTACTS

El-Ayachi Moha: moha.elayachi@gmail.com / Semlali El Hassane : e.semlali@gmail.com, Ettarid Mohamed ettaridmohamed@gmail.com / Driss Tahiri: d.tahiri@iav.ac.ma

Tel:+ 212 537 680 180 / Fax: +212 537 680 180.