BEHAVIOUR MODELLING IN GEOGRAPHIC APPLICATIONS

Graça Abrantes*, Andrea Pinheiro**
*Inst. Sup. de Agronomia/UTL - Dep. Matemática and INESC
**INESC, Centro Multimedia - Sistemas de Informação Geográfica
R. Alves Redol, 9
1000 Lisboa, Portugal

Summary

Presently, geographic applications must often support dynamic requirements. Object-oriented methods are commonly considered specially appropriate to model systems which have demanding requirements concerning dynamic. This paper focus on an object-oriented method to model the behaviour of objects in geographic applications. The proposed method was used to develop an application to assist on planning and implementing projects to install and maintain facilities' networks. This application is used to show the method advantages.

1. Introduction

The development of geographic applications with complex requirements leads to the implementation of long programs in order to support the automatic execution of several operations. The cost of the corresponding programming activities became relevant in the overall development process of geographic applications. The use of suitable methods to assist on these activities, allowing the creation of concise and easily understandable specifications, may contribute to an efficient development process. The advantages are the creation of applications developed as fast as possible and fulfilling the actual user requirements.

In the software engineer field, methodologies aim to define a correct system development process. Presently, there are several methodologies to support the development of traditional information systems. Most of them are based on data modelling. Actually, data models may constitute formal specifications of applications. The methodologies based on the object-oriented approach present several advantages. The ability to represent the static and dynamic aspects of the objects on a single data model is one of the more relevant advantages.

However, standard object-oriented methodologies cannot accommodate the specific characteristics of systems including geographic data. These specific characteristics may be supported if the standard methodologies are extended in order to provide new suitable abstractions and constructs. In these paper, we use an extension of the OMT methodology [3] to specify the static aspects of spatial objects [1] and we propose the extension of some concepts to represent their specific dynamic characteristics.

2. Dynamic aspects of geographic applications

In the past, geographic applications did not present relevant dynamic requirements. Lack of adequate technological support to such requirements can explain that situation. In geographic applications data modifications often require a long time to be completed. Usually, these changes are made in large regions and are the result of a planning process, where changes are designed, analysed and frequently rebuilt until a correct solution is achieved. Naturally, this kind of data update cannot be done in a unique and fast transaction. On the other hand, during these long transactions geographic data must remain available so that concurrent processes may execute other operations. Under these conditions, multiple access control and data integrity cannot be ensured by mechanisms like those that are used in classical database systems.

The technological improvements in the GIS field, specially the support to long transactions, has contributed to solve some of the problems that in the past constrained the development of geographic applications with significant dynamic functionality.

Long transactions support usually involves the concept of local data. The creation of local data is part of the transaction's initialisation. Basically, local data is a copy of the spatial database to work files; usually these files only include data that are located in a region selected by the user, according to each transaction needs. The updates are firstly done in those files while the database remains unchanged until the transaction finishes. When the user decides to end the transaction the work files data substitute the corresponding data in the spatial database.

The method used to support concurrent transactions depends on the decisions made in the first transaction initialisation and on the policies for concurrent access control supported by the GIS used in implementation. Broadly, these policies may go from the lock of all data involved in the transaction forbidding their use by other transactions, through the feature lock in the spatial database only when the transaction modifies that feature, to the absence of any lock (this option is generally associated with read-only transactions initialisation). As most GIS usually have a hybrid architecture, with non-spatial data being stored in a relational database, we can still distinguish several variants of the above mentioned policies, concerning the way GIS guarantee the coherence among spatial and non-spatial data.

Anyway, the data duplication in local files and the possibility of using the same data set in several simultaneous transactions, requires that the specification of the application dynamic accommodates this characteristic. In reality, that possibility means that an object can be differently visualized (i.e., can be in different states) at the same time, depending on the transaction in which the object is visualized. This situation is not common in traditional systems and this is the reason why standard methodologies do not support it.

3. Data modelling

The OMT (Object Modeling Technique) is a well-known object-oriented methodology. In this method the object model is the main product of the analysis and design stages. In the object model the relevant characteristics of objects are represented by classes, the objects' structure is represented by the attributes of the class and their behaviour by the operations of the class; the relhashionships among objects are represented by associations among classes.

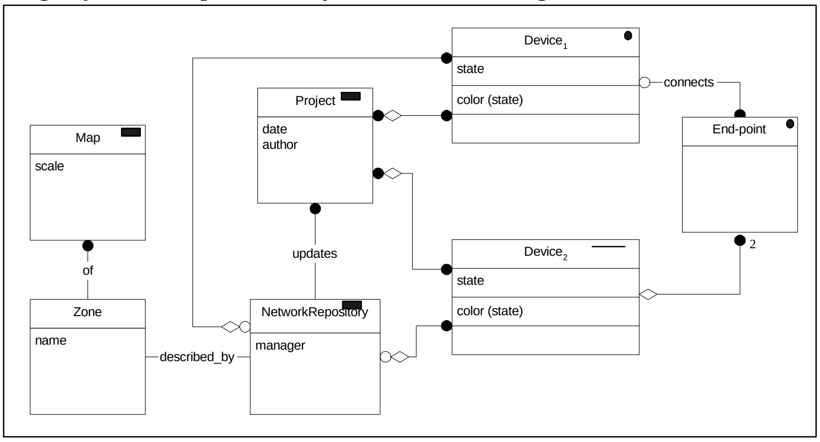

Figure 1 - Object model for a general facilities network system

Figure 1 depicts the initial object model, using the extended OMT notation, of a system for facilities network management. For sake of simplicity, many details of the system were omitted. The system requirements include several functions. This paper will only focus on the support to activities concerning the creation of new networks and improvements of already existing networks — the so-called projects. The project activities constitute long transactions.

Often, the identification of the classes' operation is not an easy activity. In difficult situations, when the objects exhibit complex dynamic aspects, it may be necessary to build models representing specifically those aspects. Based on these models, the operations of each classes can be identified more easily.

3.1 The dynamic model

The OMT methodology uses the dynamic model to describe the sequences of operations involving the objects of a given class. This kind of models is represented using the notation of Harel [2] and contains state transitions diagrams. A dynamic model uses concepts such as, state — representing objects' attribute values with specific semantics — and event — representing external stimuli that cause objects' state modification. The entry to a state may determine the execution of a sequential activity. Events and activities correspond to operations of the class that must be represented on the object model.

Figure 2 depicts the dynamic model for the classes Device$_1$ and Device$_2$. The particularity of this model is the representation of state transitions leading to two different states of a single object. The state that is visualized by the user depends on the transaction type (project design or repository management) that handles the object.

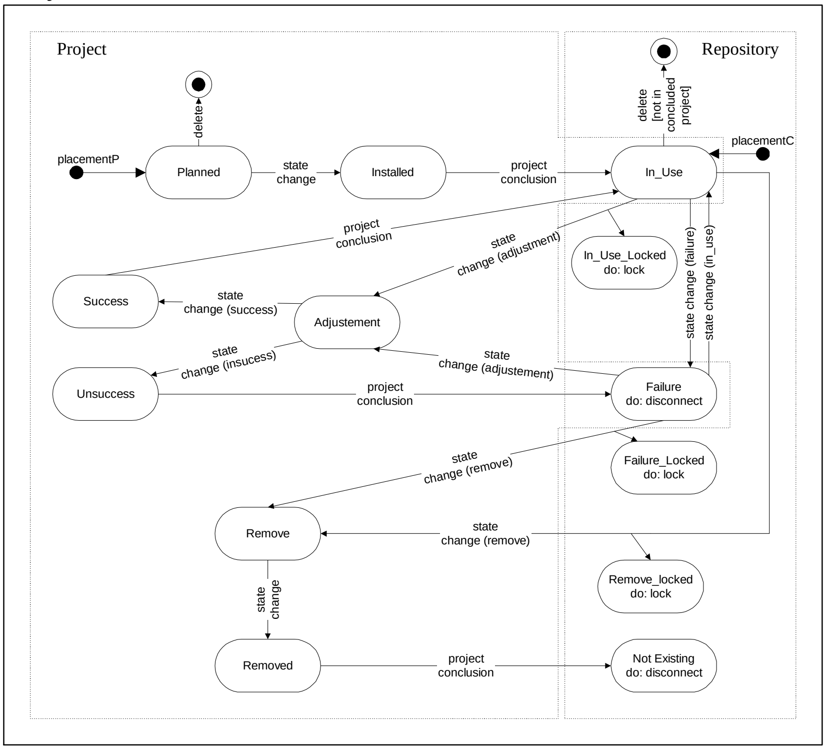

Figure 2 - State diagram for Device classes

4. Conclusion

Long transactions are a characteristic of most geographic applications and require specific techniques in order to obtain useful system specifications. The proposed technique extended the concept of state in dynamic models. One object may exhibit simultaneously two states; each one of these states is associated with one or more transaction types.

References

[1] Abrantes, G., 1996. The Object-Oriented Approach to the Specification of Geographic Applications. *JEC'96 Proceedings* edited by M. Rumor, R. Mcmillen, H.F. Ottens (Amsterdam, The Netherlands: IOS Press), pp.248-257.

[2] Harel, D., 1987. Statecharts: a visual formalism for complex systems. Science of Computer Programming, **8**, pp. 231-274.

[3] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., 1991. *Object-Oriented Modeling and Design*. (Englewood Cliffs, NJ: Prentice Hall).