EVOLUTIVE COMPLEX SCHEDULING IN INTERACTION NETWORKS FOR QUALITY IMPROVMENT IN GEOGRAPHICAL DATA BASE UPDATING

Hakima Kadri-Dahmani(1), *Cyrille Bertelle*(2), *Gérard H.E. Duchamp*(1), *Aomar Osmani*(1)

(1) LIPN - University of Paris 13 - 99, avenue Jean-Baptiste Clément - 93430 Villetaneuse - France
(2) LIH - University of Le Havre - 25 rue Ph. Lebon - BP 540 - 76058 Le Havre cedex - France

ABSTRACT

The aim of this paper is to propose a scheduling processus for the mechanism of geographical data bases updating in term of improvement of its quality. Because of the complex interacting network of the components involved in each updating cycle of this mechanism, a traditional analytic solver cannot success in most cases. So we propose to use an evolutionary computation based on a genetic algorithm which is well adapted for the complex interaction network of the involved components. A genetic algorithm may allow to improve the updating quality in increasing the validation rate of a transaction.

1. INTRODUCTION

Geomatics finds its roots in numeric geographical information defined in [1] as :

Definition 1.1 *Geographical information is the composition of a set of heterogeneous informations which are strongly linked to be able to describe the characteristic of a scene in the meaning of: localisation, geometry, topology and semantic.*

One of the most adequat support for this information is geographical data bases which have to involve, using a conceptual model, geometric data and semantic data. The Geographical Data Bases (GDB) based on vector representation are generally composed of layered classes of objects. Each layer is associated to a thematic like road tracing, fluvial tracing, buildings, vegetation,etc. The constant evolution of the real world which must be represented in such geographical data bases needs to regularly update the data. The knowledge which will serve the updating is generally associated to the concerned semantic layer. Each semantic layer has its own updating frequency which can differ from one layer to another. This heterogeneity can be the cause of some problems on the consistency maintenance of the whole GDB. In the figure 1, we describe the updating generated by a new road arrangement. The first part of this figure shows the initial map configuration. The second part shows a first uncorrect updating without

Figure 1. Updating within consistency maintenance

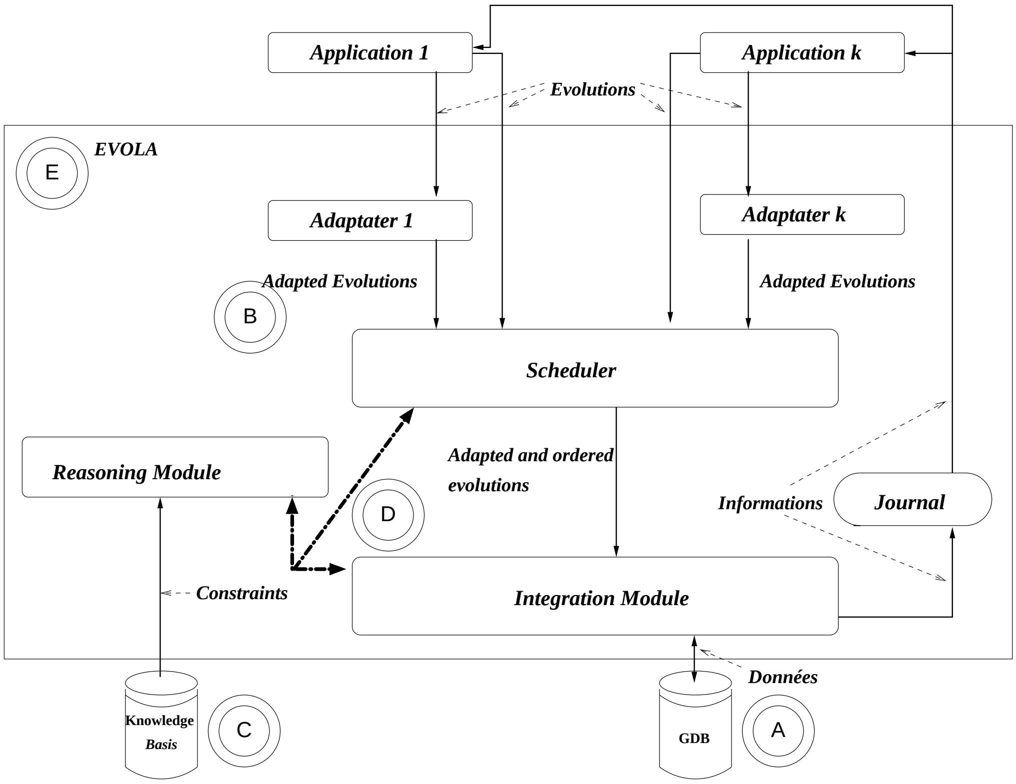

Figure 2. EVOLA architecture

checking the whole consistence, generating superposition of a new road portion over existing buildings (left bottom of the road crossing). The last part shows a correct updating after the propagation of some topological constraints between the layer of the road tracing and of the layer of the buildings position.

This example shows that a GDB must be represented with an interaction network of components to allow the right propagation of constraints within updating processes.

In [1], a system was proposed as an automatic solution to the problem of maintenance of the GDB's consistency at the time of its update. In this paper, we present briefly this system architecture. In the section 3, we locate the scheduling problem in this mechanism. Finally, in section 4, we introduce a solution that we propose to the scheduling problem during the geographical data bases updating. We also justify the advantage of this solution based on a genetic algorithm which is well adapted for the complex interaction network of the involved components. A genetic algorithm may allow to improve the updating quality in increasing the validation rate of a transaction.

2. EVOLA: UPDATING SYSTEM ARCHITECTURE

The system called EVOLA [1] has been developped in the laboratory COGIT of IGN-France (Institut Géographique National). EVOLA is a tool which allows automatic updating of geographical data bases.

The EVOLA general achitecture is summarized in the figure 2. In this figure, 4 parts are mentioned and describe the work needed to use it and the processes involved.

- The need of reorganizing the GDB to conform it to the model described previously and allowing the updating is mentionned by the part (A).
- The part (B) (resp. (C)) mentions that we need to model the updating informations (resp. the constraints representations) as previously described to adapt them to the system.
- The part (D) corresponds to the updating processus which is composed of a 3-modules kernel: *the reasoning-based engine, the integration module, the scheduler*.

We can see in Figure 2 how the different components do interact during the updating integration processus. First of all, the system receives an external set of updating operations from an external application. These evolutions may be expressed in the application format or in EVOLA format. If the first case, the set of evolution is passed to the *Adapter*. This last one converts it in the EVOLA format, corresponding to a set of specifical canonical operations (creation, supression, descriptive modification and identity). Then the *sheduler* computes the best order to execute in the right sequence the whole set of these canonical operations. When the *Integration Module* receives the set of evolutions, it has to execute the different updating operations included in this set. This execution must

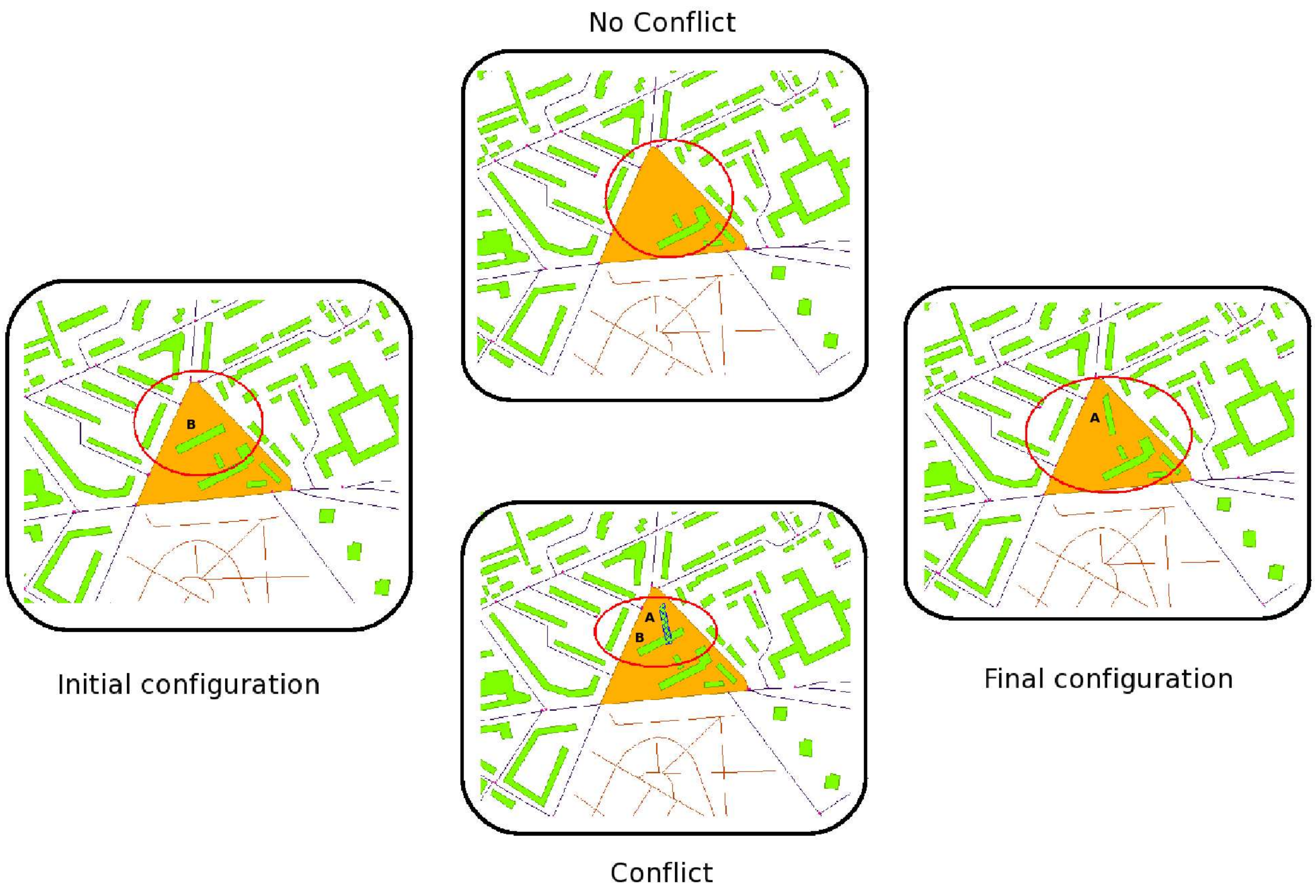

Figure 3. Scheduling Problem

be free from conflicts otherwise the Integration Module has to detect them and to take decisions as regards these conflicts. To detect the conflicts, it uses the *Reasoning module* which, basing itself on the *constraints database* and the algorithms of consistence preservation, decides of the manner of modifications integration on the data of the database. at the end of each sequence of operations, a report of the various treatments of the operations is saved in the *Journal*. This report specifies for each operation, if it was carried out successfully or not. In the case of failure of the operation, the journal specifies the kind of conflit generated by this operation and the constraints which was violated.

3. SCHEDULING PROBLEM

The solution proposed in [1] to realize updating of geographical data bases and to check their consistency consists in the use of incremental processing. Each step of incremental processing consists in (i) the execution of a canonical operation and (ii) the propagation of its influence on the whole geographical data base and (iii) checking consistency of the geographical data base. ....which relies on the fact the rate of geographical data base inconsistency, at the time of its update, strongly depends on the order of execution of the updating operations.

Consider the representation of the geographical scene in a geographical data base, corresponding to the part ”initial configuration” in the figure 3. Consider that a user of this geographical data base needs to update it because of changes in the scene. The modification consists in the creation of the building A and the deletion of the building B. The part ”Conflict” of the figure 3 shows that the addition of the building A in the database before the deletion of the building B, generates conflict. This conflict lies in the building A passing over the building B. Depending on the kind of the conflict, the whole input updating sequence may be cancelled. In the other case, the input updating sequence processus will continue and the conflict may disappears later (figure 3, part ”Final configuration”)., i.e. during the execution of an other updating operation from the sequence (the deletion of the building B). Although the conflict may disappears in another cycle of updates, it needs a specific treatment at the time of the cycle during which it was created. Elsewhere, if the sequence order differs: if the user starts by deleting the building B, as shown in the part ”No Conflict” of the figure 3, and then creating the building A, no conflict is generated (figure 3, part ”Final configuration”).

The example shows that the execution of the second sequence (the deletion of B before the the creation of A) is less expensive than the execution of the first sequence (Tthe deletion of B after the the creation of A). Moreover, this order allows the validation of the updating when the second one may cancel it.

The goal of the scheduler defined in the EVOLA architecture is to compute the best order of the operations inside the input sequence.

4. EVOLA: UPDATING SYSTEM SCHEDULED BY GENETIC ALGORITHM

The scheduler has to minimize the apparition of conflicts. A conflict consists in the violation of a constraint. A constraint is defined on some relations between GDB objects. In the EVOLA tool, these relations are essential for the constraints propagation descriptionand can be represented with an interaction network over the GDB elements. We divide these relations in two main parts:

1. *Semantic relations*: They are composed of *composition relations* which allow to describe aggregation of some objects and *dependance relations* which describes the fact that the modification of one object can lead to the modification of other ones;

2. *Topological relations*: they describe the type of topological connection between objects. We propose to use the 9-intersection model from Egenhofer and Herring [2].

The interaction network is, in that way, built with the support of a graph of relations. As the updating is based on a propagation mechanism as described in [1], the graph will structurally evolve during the computation. This dynamic evolution will generate some irregularities in the function we have to minimize and which computes the number of conflicts. This kind of problems is well-known to be strongly solved with classical optimisation methods, especially with analytical ones, and genetic algorithms are a good alternative processes to solve them.

The genetic algorithm proposed here consists to define a population of individuals which are all potential solutions for the scheduler. These solutions are represented as chromosoms constituted as the sequence of all the canonical operations given to the system by the adapters. Each canonical operation is an allel and we use the 3 classical genetic operators: duplication (with selection processus), cross-over and mutation. The fitness function has to minimize the apparitions of conflicts and so to maximize the validation rate for all the operations in a sequence. A canonical operation is validated if it doesn’t generate any conflict.

5. IMPLEMENTATION AND EXPERIMENTS

The whole system has been originaly developed in the COGIT laboratory where it has been implemented. This system is in operational practice and has been connected to the framework OXYGENE [3] of this laboratory. A methodology for its validation has been developped and has proved that the mechanisms are efficient, even if some rejections could be avoided with a better scheduling. The genetic algorithm engine is still under development and validation and it will improve the quality of the whole system.

An experiment has been developped on the IGN GDB concerning the Angers French town zone. Using some matching technics between two of these GDB from 1994 and 1996, we have built a set of updating informations. A *precision* indicator is computed as the rate between the right decisions and all the decisions taken by the EVOLA system. From a specific experiment based on 30 canonical operations, we have obtained a precision indicator equal to 0.948 which is sufficient to validate the whole processus. A detailed description about validation can be found in [1].

6. CONCLUSION

This paper has described an effective processus for the updating of geographical data bases. We have presented how the consistency of the GDB can be violated if a wrong scheduling is computed. A whole architecture explaining the place of the scheduler in the incremental process, has been shown. The automatic updating proposed is based on a propagation method. This method leads to a dynamic evolution of constraints formulation between the GDB elements. We propose a genetic algorithm to implement the computation that the scheduler has to do to improve the validation rate of each updating transaction. Satisfying results have been obtained in some experiments made on IGN geographical data bases, even if the genetic computation is still in progress.

7. REFERENCES

[1] Kadri-Dahmani, H., *Mise à jour des Bases de Données géographiques et maintien de leur cohérence*, Ph.D. thesis, University of Paris 13, 2005.

[2] Egenhofer, M. and Herring, J. R., “Categorizing Binary Topological Relations Between Regions, Lines and Points in Geographic Databases,” Technical Report, Departement of Surveying Engineering, University of Maine, 1991.

[3] Badard, T. and Braun, A., “OXYGENE: an open framework for the deployement of geographic web services,” in *Proceedings/Actes, 21th International Cartographic Conference, Durban, South Africa*, 2003.