Cartographic database updating

Cécile Lemarié and Thierry Badard
Institut Géographique National, Service de la Recherche
2 à 4, avenue Pasteur - 94165 Saint-Mandé Cedex - France
Email: {Thierry.Badard ; Cecile.Lemarie}@ign.fr

Abstract
IGN (the French National Mapping Agency) derives cartographic products from reference geographic databases through complex processes, including among others, generalisation and label placement. Derivation is partly automated but still a long interactive part is necessary to achieve a good cartographic result. Now that cartographic derivation processes are rather well known and reference databases are updated regularly, the next step is to be able to rapidly update these maps, without performing the whole cartographic derivation process anew but only by using updates which have been captured in the reference databases. This paper reports on research, development and tests carried out by the Research Department of IGN to automate the updating of cartographic databases by using updating information coming from their reference databases.

1 Introduction

IGN has decided to produce a new 1:100.000 scale map series named Top100 by deriving it from one of its reference database: the BDCarto® (a 1:50.000 database with 10-meter resolution produced by the IGN). This project (named "Carto2001, Cartographic Space Odyssey") started in June 1999 and aims at defining a production line compliant with an automated updating of the maps. The project is responsible for the definition of both the derivation process which will produce the initial map and the updating process of this new map. It's the first time at the IGN, that a new map series takes into account its updating process so early in its design.

After a brief presentation of IGN researches in both updating and generalisation areas, the next subject discussed is how the cartographic database must be designed to support automated updating. Up to now, the design of the cartographic database was often made independently of the updating process which was studied later. Practice has demonstrated that such a method led to an insufficiently automated updating process, mainly because information may be lost or become inconsistent. So, the IGN "Carto2001 Project" has in charge of defining both the derivation and the updating processes to be sure that the map will be as automatically "updateable" as possible. The paper details the rules which have to be taken into account while designing of the cartographic database.

Finally the paper presents the updating process defined and tested on the Top100 series. The four main steps of the updating process will be explained: *selection* which consists in choosing to propagate in the cartographic database only the updates which have a meaning for the map or to preserve its consistency. *Automated update integration* which propagates the effects of the updates in the cartographic product, taking into account generalisation rules, symbolisation, cartographic objects (labels, kilometres pointers…). *Interactive update integration* for updates that failed to be automatically integrated. *Validation* which checks the cartographic database to detect inconsistencies or missing updates which have not been integrated. The four steps will be illustrated by results provided by first Top100 production tests. First results and future works will then be presented.

2 Context

The "Carto2001, Cartographic Space Odyssey" project has been launched after ten years of research in the generalisation and updating domains. So, it benefits from a strong experience and can be considered as one of the first industrial implementations of these two research areas.

2.1 Generalisation: last progress

2.1.1 AGENT European Project

After ten years of research in cartographic generalisation, IGN has engaged the three years AGENT European project. It finished in December 2000 and was gathering five partners: the COGIT laboratory of IGN, LaserScan Limited (editor of the LAMPS2 GIS), the INPG Leibniz laboratory (specialised in multi-agent techniques), and the Geography Departments of Zurich and Edinburgh Universities.

The purpose of the project was to design a prototype for cartographic generalisation using multi-agent concepts and its implementation in a commercial GIS software [Lamy et al., 1999]. Moreover, the prototype had to integrate generalisation algorithms developed in three research laboratories (COGIT, Zurich and Edinburgh).

Principles involved in the AGENT project are mainly based on the PhD research works of Anne Ruas [Ruas, 1998]. Automatic generalisation consists in working with cartographic objects which have to resolve numerous conflicts (with themselves but also with other objects) in order to produce legible maps. AGENT's method aims at giving them autonomy to solve such conflicts by themselves. These objects are then called agents and they have a set of goals to achieve. They are able to analyse their state (some measures are available), to trigger algorithms and to assess if conflicts are solved. The state of each object is defined by a set of constraints to satisfy, which translate the cartographic rules to take into account in the derivation process.

These principles have been implemented in the LAMPS2 software (Laserscan Limited) and constitute the bases of the generalisation prototype involved in the Carto2001 project.

2.1.2 LAMPS2 software

Results of the AGENT project seem to be very powerful for the automation of the Top100 derivation process. That is mainly why LAMPS2 has been naturally chosen as GIS platform for the Carto2001 project, but the software provides also other very interesting features for the development of the updating prototype:

- LAMPS2 manages updates by creating versions, so during an updating process it is possible to query the database not only in its current version but also in its older states. It is then very easy for example to cancel an update and retrieve the initial state of an object.
- LAMPS2 relies on a robust et efficient Database Management System (DBMS) able to store and handle the whole French territory in a single database. It then avoids the management of one database by map, which imply to update several times the same part of the database to ensure its consistency.
- LAMPS2 implements Object Oriented concepts which allow users to easily define the behaviours of an object. This aspect has especially been enhanced in LAMPS2 by the mean of a peculiar type of methods (named reflex methods). Such methods may automatically be triggered by the system when objects are involved in a process. This allows a very easy definition of the interactions between objects (especially for the updating process).
- LAMPS2 can be used in a multi-users environment which makes the simultaneous updating of different areas possible.

2.2 Updating researches

In 1996, IGN via the COGIT laboratory has decided to engage researches in order to address the propagation of updates in multi-scale geographic databases. Results of this research are detailed in the PhD research work of Thierry Badard [Badard, 2000]. Nevertheless, main innovations concern:

- The implementation of a generic method for the retrieval of updates between two version of a same geographic database. This method relies on geographic data matching algorithms and allows for the extraction of a minimal but detailed updating information, close to the modifications performed in GIS.
- In order to deliver this updating information to users, new delivery modes for updates have been defined. One of these modes relies on the XML (eXtensible Markup Language) encoding of the updating information [Badard and Richard, 2001]. It allows a minimal and dynamic description of updates and enables such a transfer on networks such as the Internet.
- A mechanism for the integration of updates and the propagation of their effects in multi-scale geographic databases (which includes systems with user added information) has also be defined and tested. It relies on a rule base which enables to automate and control the propagation of updates in order to preserve the consistency of databases and avoid information losses. About 85% of the updates in different tests have automatically been propagated by this mechanism.

The Carto2001 project will be in IGN the first product which will take advantage of these innovations: the structure of the updating process and updating data involved in the project are inspired by these research works.

2.3 The Carto2001 project's challenge

Up to now, IGN researches have separately considered the updating and generalisation problems. Updating researches have especially focussed on the updating of geographical databases or cartographic databases for which the degrees of generalisation were very close. The Carto2001 project has to move a step forward: updating a cartographic database where generalisation is significant.

Updating such data implies to define new updating rules (since modifications in the reference database induce cartographic evolution in the Top100 database) and to take into account the propagation of updates on cartographic objects which are not necessarily captured in the reference database (e.g. labels, bridges, kilometres…).

For example:

- A change in the geometry or semantics of an object may result in updates for the cartographic product. Such evolutions involve for instance displacement or bend generalisation algorithms due to an increasing of the symbol width.
- Cartographic objects are linked to each other and the update of an object often involves other objects to be modified (e.g. a change in the geometry of a road may results in displacements of road numbers or in orientation modifications of the services symbol connected to this road).
- Displacements between objects in the Top100 and in the reference database can be important but both topological and geometrical relationships have to be preserved during the updating process.

3 Architecture of the Carto2001 prototype

It can be divided in three main sub-prototypes:

- A generalisation prototype (relying on the AGENT tools) which enables the derivation of the first version of the Top100 from BDCarto®.
- A label placement prototype which is based on a independent software and allow for the automation of names and symbols placement on the map.
- An updating prototype which has been implemented in LAMPS2 and aims at performing a propagation of the updates from the reference database (BDCarto®) into the Top100 with a high level of automation.

Despite they are separately presented, different interactions exist between these 3 sub-prototypes and are pointed up in the next sections.

3.1 Design of the cartographic database

As mentioned before, the main goal of the project is to automate the updating of a cartographic database by using evolution data. Such a goal can only be achieved if a great attention is drawn during the design and modelling processes of the first map. Previous experiments have indeed failed because constraints induced by an updating process relying on evolution data have not sufficiently been taken into account during these steps. These rules have been applied to the Top100.

The delivery of updating information relies on the delivery modes defined in [Badard and Richard, 2001] and base on XML. It corresponds not to a complete delivery of the updated objects: only the updated parts of the object are delivered in their new version (for example only an attribute or only the object geometry). This delivery method implies to define the following constraints:

- objects have to be identified by a key attribute which must be unique and unaltered in the cartographic database
- all the operations concerning the cartographic objects are not allowed: for instance, the object aggregation is not possible in the cartographic database because it requires a complex system allowing to trace the splitting and merging operations (to know when an object is updated which aggregated object it corresponds).

In order to manage the propagation effects of updates on cartographic objects not captured in the reference database (e.g. labels, bridges…), it is necessary to properly define the “correlations” (i.e. all the relationships allowing to perform a consistent and complete propagation of updates [Badard and Lemarié, 1999]) between the cartographic and reference objects. These correlations can be stored in the cartographic database or retrieved depending on the difficulty to establish them.

It has also been decided to store the generalised objects as well as the reference ones in the cartographic database so that when an object is updated, it can be displaced in the same way as its neighbours were during the generalisation process.

3.2 First map derivation

As the updating process will have to perform the same type of operations on the updated objects as those involved in the derivation step for the first version of the Top100, architecture of the first map derivation process is briefly described in this section. It may be divided in three main parts (see Figure 1):

- **data import:** this step consists in importing in LAMPS2 the reference data from BDCarto® and assigning them their cartographic symbols.

- **data generalisation:** once data are imported and symbolised, it is necessary to generalise them. Two types of generalisation operations will be performed on geographic objects: **intra-object generalisation** consisting in solving conflicts due to the shape of the object (bend removal, smoothing, shape exaggeration…) and **inter-object generalisation** which aims at solving conflicts due to the relative position of objects (displacement, road interchange processing…). These two types of conflict will be in fact managed in parallel by the AGENT prototype and even if it has not yet been implemented into the Carto2001 project, the generalisation process will likely to comprise two steps: **an automatic step** (which should be able to solve all inter-conflicts and a great part of the intra-object conflicts) and **an interactive part** where conflicts should be pointed out by the AGENT system. All the generalisation conflicts are detected but in particular cases AGENT prototype is not able to determine by itself which tool is the most suitable to solve the conflict. Users would have to take the decision and will be provided with a set of tools in order to interactively achieve this task.
- **label placement:** once all data are correctly represented on the map, it is necessary to place their names and labels. This placement is performed by using a specific software dedicated to label positioning, which has been developed in the Carto2001 project. It uses a specific image of the map where each pixel represents the degree of acceptance for a label to overlap this position. This software should be able to automatically place more than 85% of the labels. Labels which have been correctly positioned are then imported in LAMPS2 and remaining labels are then interactively processed by using specific tools available into the Laserscan GIS.

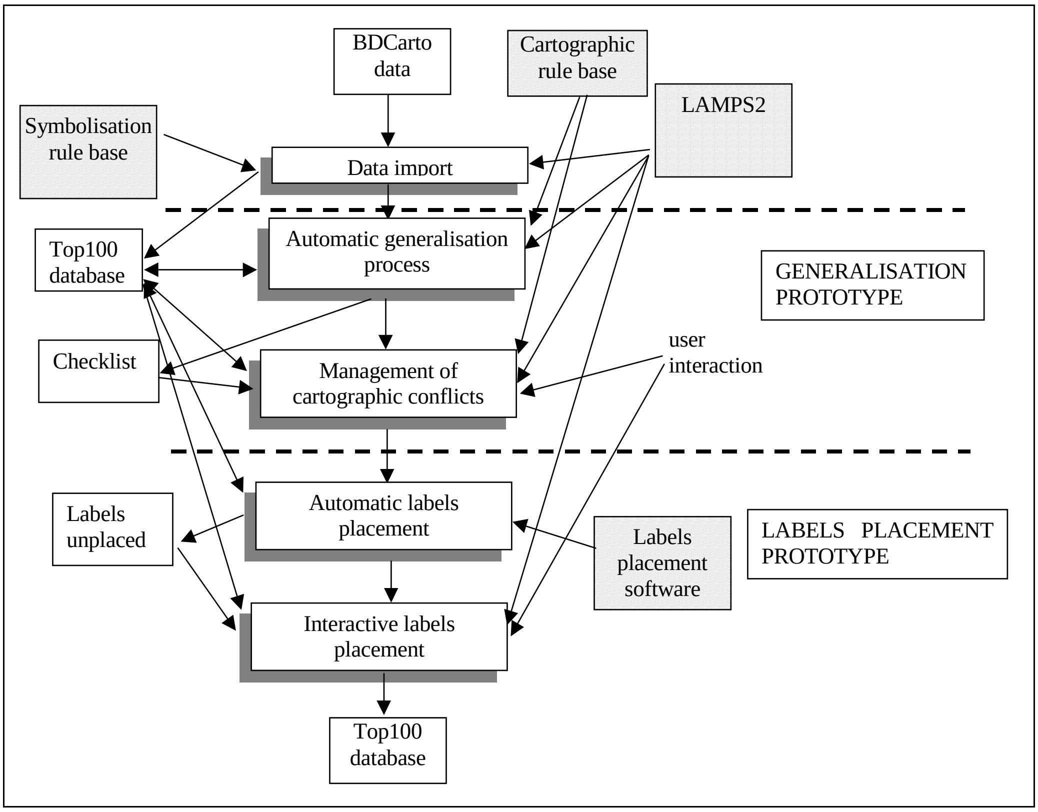
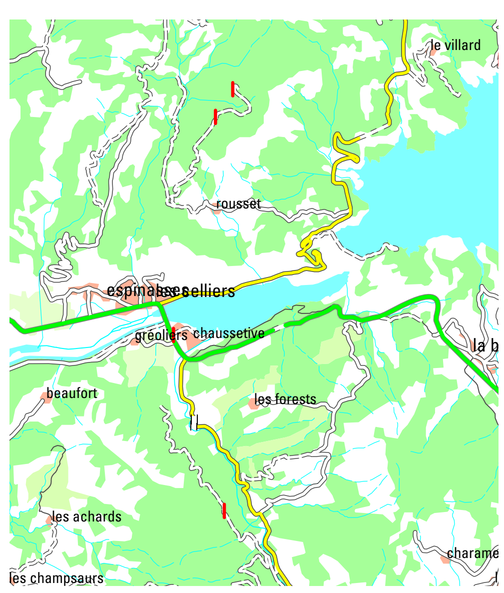


Figure 1: Two steps derivation of the first map

This derivation process is very expensive and is estimated to 8 months for each map, which makes useful an efficient updating process. The following figures illustrate on a small area the differences between data resulting from the first step (i.e. after import and symbolisation) of the derivation process and data as they should be in the final map.



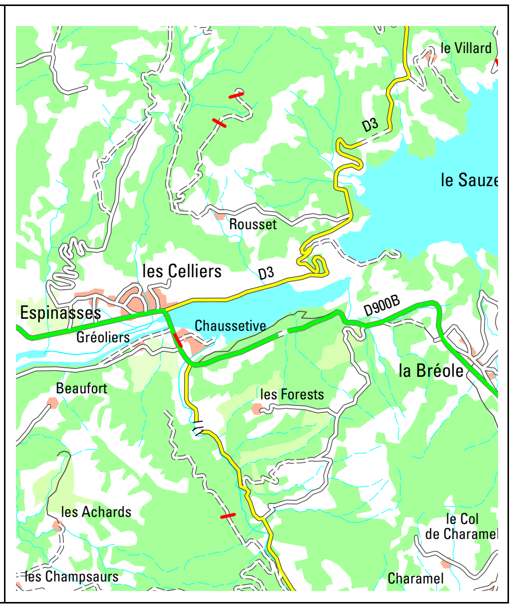


Figure 2: Symbolised BDCarto® data **Figure 3: Final Top100 map**

3.3 Updating process

The updating process has been developed in LAMPS2 and can be decomposed in two main parts:

- an updating engine which implements a set of functions called in a predetermined order
- an updating rule base which allow for the definition of behaviours to adopt when a particular operation is performed on a specific object.

The updating engine enables the integration of updates and assumes that objects are modified without creating any conflicts in the database and without altering all the topological constraints. The updating rule base deals with the cartographic aspect of the database, by controlling the effects of the updates on the cartographic objects. There are obviously various interactions between these two components, but they are mainly managed by the LAMPS2 engine. Each time an object is modified by the updating engine or by users, LAMPS2 automatically refers to the reflex methods (call-back mechanism) which implement the updating rules.

The updating engine may be decomposed in four main steps (see Figure 4): filtering, automatic updating, conflicts solving and consistency checking. It is connected to the generalisation prototype in order to perform generalisation processes on the updated objects if necessary. Connection to the label placement software has not been implemented yet. It is indeed not really adapted to local placement of labels and it seems for us easier compared to the number of labels which should be updated to integrate it in the updating rules methods for detecting label placement to modify or to create (road number for example).

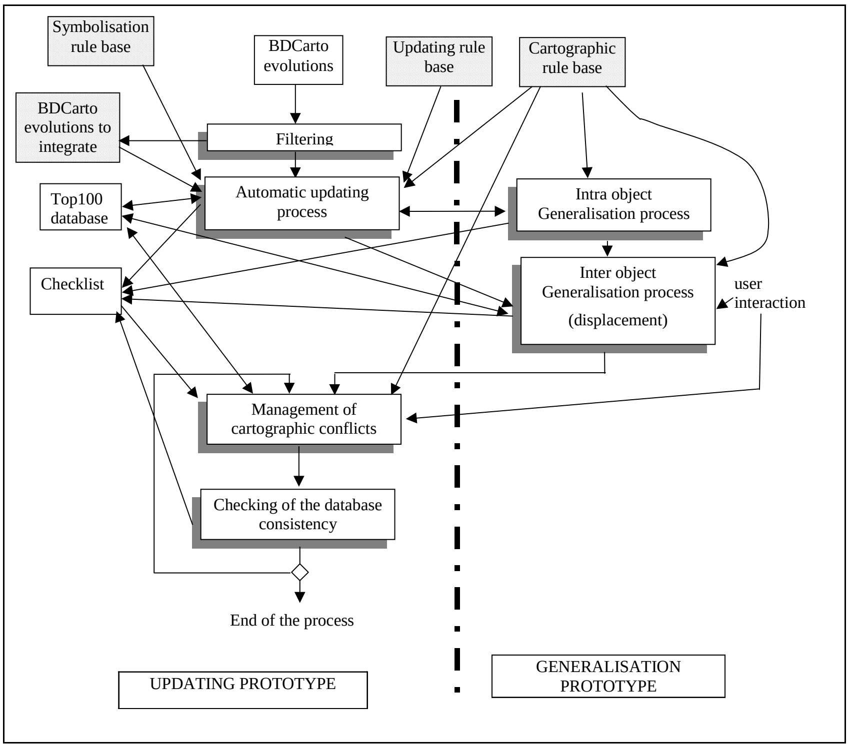


Figure 4: Overview of the updating process architecture

3.3.1 Filtering step

The cartographic database is a distorted image of the reference database, and some modifications performed in the reference database are not always significant for the map. In this step, BDCarto® evolutions are analysed: updates which are not "cartographically" significant are ignored and the other ones are imported in the database (in particular classes). Criteria involved in this filtering step are for example: minor modification of the geometry (i.e. less than a threshold), updating of insignificant attributes. At this step of the updating process, about 30 to 50% of the geometrical updates can be filtered.

3.3.2 Automatic updating

This step constitutes the kernel of the process and consists in the integration of updates and propagation of their effects in the cartographic database. Updates are processed in a defined order to avoid incoherent state: for example, deletions are integrated before any other updates. For each update, geographic objects involved in the modification and methods for maintaining (at each step) the consistency of the database are identified. Updating rule are then automatically called by LAMPS2 to define the effects of the update on the cartographic aspect of the object (symbol change, symbol orientation) and on the other surrounding objects (road number, bridges, kilometres…). Propagation of updates on surrounding objects can not always be defined during the updating of an object since the context surrounding the object is not up-to-date, so for specific treatment (road numbering for example where the number is placed considering a set of road sections) the effects of updates are not calculated during the integration step but at the end of the process (when data directly corresponding to the reference database are up-to-date).

The two main characteristics of the process are:

- The consistency of the database is always maintained. It is important to mention that the BDCarto® stores graph relationships which indicates the connections of each road sections and that the evolution data also delivers the modifications concerning these relationships. They can then be used to maintain the connectivity of the network. These evolutions are used to ensure that an updated object is correctly connected to an existing part of the network. An evolution can thus be integrated in the cartographic database only if one of the object connected to the updated object is already in the cartographic database and up-to-date. This constraint is the easiest way to always preserve a correct network topology.
- The previous generalisation process is used for integration of updates. To avoid local distortion and topological errors, it is necessary to take into account the existing displacement BDCarto®/Top100 to reproduce it on the updated object. Figure 5 illustrates the results provided by the updating process without taking into account this constraint (blue lines represent the geometry of updated BDCarto® objects). Figure 6 demonstrates this constraint enables a consistent network updating without shape alteration and topological errors.

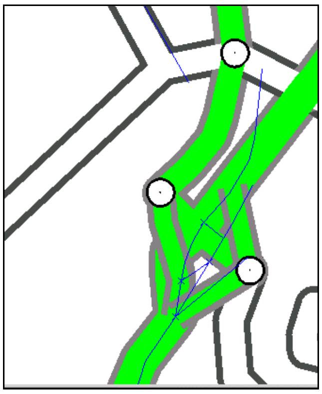


Figure 5: updating whithout taking into account previous displacement

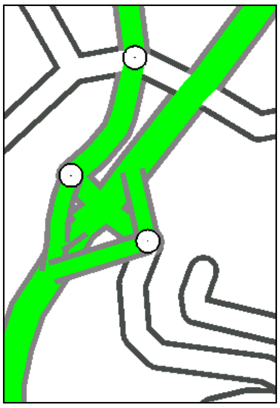


Figure 6: correct updating

3.3.3 Conflicts solving

All the propagation effects can not fully be defined by the way of the updating rule base, so during the automatic step, objects which must be updated or checked by the operator because the updating process can not automatically integrate them, are "marked" and a description of the problem is attached to the object.

The main point in this interactive step is that all the updating rules are still automatically triggered by the LAMPS2 engine, so it is important to mention that there is no difference between cartographic processing of objects which are updated by the automatic process and objects which are updated by the operator.

3.3.4 Consistency checking

This step checks that the updates have been correctly integrated and propagated: all the evolutions have been taken into account, topology is in accordance with BDCarto®, etc. the versioning mechanism of LAMPS2 is then used to check that no unfortunate updates have been performed (deletion of an object which has not been removed in the BDCarto®, modification of the object identifier) in order to enable future updating.

4 Results and outlooks

At present, a great part of the updating process has been implemented and tested. Except the connection to the generalisation prototype (as it is not yet fully defined), the updating engine is implemented. The updating rules have been defined and enable the updating of the road network taking into account labels overlapping and cartographic objects created for the map: bridges and road numbers.

A first experiment has been engaged on a map sheet to assess the efficiency and the robustness of the updating process before to take into account other themes. The area considered for the test was very interesting since the reference database had undergone a lot of modifications.

Since the label placement prototype and the generalisation prototype are not yet fully developed, it has been decided to generate the first map by using research tools which should give the same result as the Carto2001 future tools.

The following table indicates the number of updates which have to be considered on the test area.

Updating type	Number of updates
Creation	1328
Destruction	824
Semantic modification	416
Semantic and geometric modification	127
Geometric modification	44
Aggregation	170
Total	**2913**

The updating process has been launched and its automatic part lasts about an hour. It achieved to integrate all the updates in the cartographic database. Some labels have had to be checked (identified by the system) since the process is not able to compute all the label placements (about 40 labels and 40 roads to check that they didn't need to be numbered). This interactive part is estimated to less than 3 hours because LAMPS2 tools makes possible to guide it. At present, the generalisation is not taken into account in the prototype and will certainly involve a part of interactive work which should be relatively insignificant, as updated objects are connected to the existing network which has already been generalised

The following images figures illustrate a small area before updating (Figure 7) and after updating (Figure 8), labels to check have been represented with a purple point.

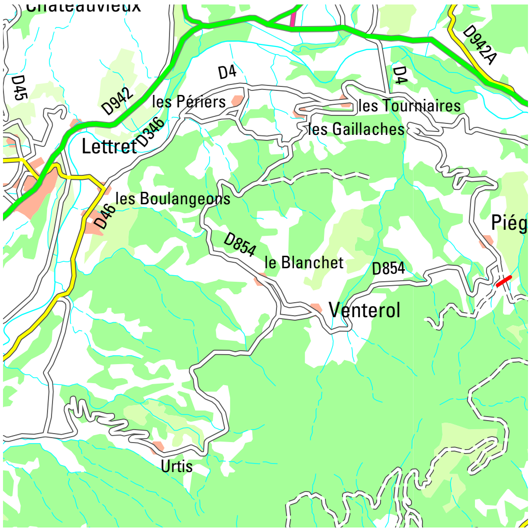


Figure 7: before updating

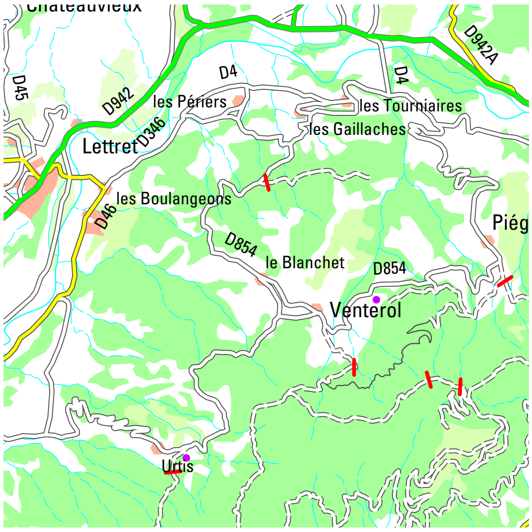


Figure 8: updating result

5 Conclusion

The cartographic updating process implemented in the Carto2001 project is the concrete result of IGN researches in generalisation and updating. The architecture of the process makes its adaptation to another map derivation possible. It requires only the enhancement of its updating rule base. The next step will be its connection to the generalisation prototype and should confirm the efficiency and robustness of such a tool.

The "Carto2001, Cartographic Space Odyssey" has fully satisfy one of its main constraints: significantly decrease the updating costs. The first test has demonstrated that automatic updating should considerably reduce delays and costs. We expect to drastically decrease them to about 10% of the current updating delays and costs.

6 References

Badard, T. and Lemarié, C. (1999). Propagating updates between geographic databases with different scales. Chapter 10 of Innovations in GIS VII: GeoComputation, Atkinson, P. and Martin, D.(Eds.), Taylor and Francis, London.

Badard, T. (2000). Propagation des mises à jour dans les bases de données géographiques multi-représentations par analyse des changements géographiques. Mémoire de thèse de doctorat en Sciences de l'Information Géographique de l'Université de Marne-la-Vallée, Marne-la-Vallée, France, 115 pages.

Badard, T. and Richard, D. (2001). Using XML for the Exhange of updating information between geographical information systems. Computers, Environment and Urban Systems (CEUS), vol. 25, Elsevier Science Ltd., Oxford, pp. 17-31.

Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W. A. and Weibel, R. (1999). The Application of Agents in Automated Map Generalisation, Proceedings of the 19th International Cartographic Conference (ICA'99), ICA/ACI (Eds.), Ottawa, Canada, pp. .1225-1234.

Ruas, A. (1999). Modèle de généralisation de données géographiques à base de contraintes et d'autonomie, Mémoire de thèse de doctorat en Sciences de l'Information Géographique de l'Université de Marne La Vallée, Marne-la-Vallée, France.