Collaborative Generalisation: Formalisation of Generalisation Knowledge to Orchestrate Different Cartographic Generalisation Processes

Guillaume Touya, Cécile Duchêne, Anne Ruas

Collaborative Generalisation: Formalisation of Generalisation Knowledge to Orchestrate Different Cartographic Generalisation Processes

Guillaume Touya[1], Cécile Duchêne[1], Anne Ruas[1]

[1] Laboratoire COGIT, IGN, 73 avenue de Paris, 94165 Saint-Mandé Cedex France
{guillaume.touya, cecile.duchene, anne.ruas}@ign.fr

Abstract. Cartographic generalisation seeks to summarise geographical information from a geographic database to produce a less detailed and readable map. This paper deals with the problem of making different automatic generalisation processes collaborate to generalise a complete map. A model to orchestrate the generalisation of different areas (cities, countryside, mountains) by different adapted processes is proposed. It is based on the formalisation of cartographic knowledge and specifications into constraints and rules sets while processes are described to formalise their capabilities. The formalised knowledge relies on generalisation domain ontology. For each available generalisation process, the formalised knowledge is then translated into process parameters by an adapted translator component. The translators allow interoperable triggers and allow the choice of the proper process to apply on each part of the space. Applications with real processes illustrate the usability of the proposed model.

Keywords: cartographic generalisation, constraints, ontology, interoperability.

1 Introduction

Cartographic generalisation is a process that seeks to summarise and characterise geographical information from a geographic database in order to produce a less detailed and readable map. Automatic generalisation processes were necessary to ease the production of map series and are growingly required nowadays with the development of on-demand mapping. Many automatic generalisation methods were developed in the past years but none is actually able to tackle all the problems raised by thematic and landscape heterogeneity present in a map [1]. Rather than developing another process that would try to solve all problems of the generalisation of a map, we believe that trying a collaborative approach is a better solution. The aim of this work is to make the available generalisation processes collaborate by generalising only the part of the map they are good at. To simplify, we want to know when, where, how and why to apply a generalisation process. When developing the first generalisation processes, research already tackled these questions concerning the sequencing of atomic algorithms [2], but the problems raised are quite different at the process level.

The paper deals with one aspect of the solution of the generalisation process sequencing problem: making different generalisation processes interoperable to be sequenced in neighbouring or identical part of cartographic space may be difficult. For instance, in the agent-based process of [3], cartographic constraints, that express the map specifications, are translated in objects with methods to monitor the process, in the least squares process of [4], the constraints are translated into equations to monitor the process and the road selection process of [5] is monitored by a big set of parameters. In order to deal with this heterogeneity of inputs and make the processes interoperable, we propose to formalise specifications and cartographic knowledge as (1) constraints, (2) rules sets, (3) ontology and (4) process descriptions.

The second section of the paper presents briefly the collaborative generalisation model we propose to optimise the sequencing of generalisation processes. The third section deals with the formalisation of the cartographic knowledge to enable the collaboration between processes. Then, the fourth section explains how the formal knowledge is used to orchestrate the processes. Finally, the last section draws some conclusions and details ongoing work on the proposed collaborative model.

2 A Collaborative Approach for Generalisation

2.1 Definition of Collaborative Generalisation

We define collaborative generalisation as an approach that makes generalisation processes collaborate to generalise the part of the space they are relevant for (Fig. 1). The data to generalise is partitioned in spaces adapted to the available processes (urban areas, rural areas, mountain areas and road network in Fig. 1).Then, each space is generalised by the most appropriate process, the mapping being guided by knowledge on automated cartography, on the user specifications and on the processes capabilities. The side effects at the generalised parts neighbourhood are monitored along the whole collaborative process. Indeed, if "process 4" of Fig. 1 displaces the road network after the other three processes were triggered, it may cause new overlap conflicts with already generalised buildings and such conflicts have to be corrected.

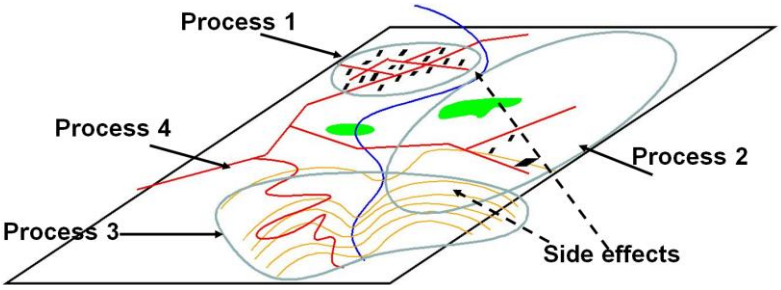


Fig. 1. The collaboration principle between generalisation processes. A process 1 is carried out on the town area, etc. Side effects are corrected at the neighbourhood of application spaces.

We choose the notion of *Collaboration* in analogy with Multiagent Systems where there is collaboration when the agents share a common goal and coordinate to achieve it [6]. We consider that the Collaborative Generalisation approach makes the processes collaborate to reach the common goal of a well generalised map.

2.2 The Issues Related to Collaborative Generalisation

Several problems are raised by the collaborative generalisation approach. The first question concern the partitioning of the space into portions relevant for a generalisation process that we call geographic spaces: it is necessary to define the relevant spaces for each available process, their relevant boundaries, and to develop algorithms to create automatically the outline of the space. Moreover, such an approach requires to model what happens at the boundaries of the generalised geographic spaces: side effects have to be monitored. It is also necessary to find a method to reach the relevant sequence to apply.

Furthermore, manual and automated cartographic generalisation require treatment homogeneity over the map. The use of different processes to generalise a complete map could jeopardise homogeneity so the collaborative generalisation approach has to take care of this issue.

Finally, some problems of collaborative generalisation are due to the use of different processes that were not developed for working together. This issue is close to the problem of designing a generalisation process based on web services [7]. Thus, it is necessary to know how the underlying model can enable the sequencing of processes with different inputs and outputs. [8] proposes a method two combine three generalisation processes into one model and highlights the issue interoperability between generalisation processes modelled differently.

2.3 Necessary Components for a Collaborative Generalisation Approach

We propose to divide the collaborative generalisation approach in five main components and three main resources (Fig. 2): the *partitioning*, *side effects*, *scheduling*, *registry* and *translator* components and the *geographic spaces*, *formalised generalisation knowledge* and *available processes* resources. This subsection describes and illustrates these components and resources. We define a resource as the required elements that can be considered as inputs of the generalisation as they are used by the Collaborative Generalisation process or guide it. We define a component as an element that is acting in the Collaborative Generalisation and that uses resources as inputs and outputs.

The *available generalisation processes* are the generalisation processes that are accessible from the software platform where the Collaborative Generalisation model is implemented. The processes can either be implemented on the same platform as the model or called as web services, as in [7].

The *geographic spaces* are the portions of initial data that are relevant for generalisation processes and that help to process large amounts of data [1]. These spaces can be metric (i.e. a limited part of earth) as the urban or coastal areas, thematic as the road network (relevant space e.g. for the elastic beams [9]) or mixed as the mountain roads. The geographic spaces do not necessarily form a partition and often overlap as a rural and a mountain space.

The *partitioning* component is composed of spatial analysis algorithms capable of delimiting the spaces as in [10]. The partitioning component allows creating the relevant geographic spaces at the beginning of the collaborative process. The partitioning component notably requires to know which are the spaces that are useful to computed according to the user specifications and the available processes. Such knowledge is included in the *formalised generalisation knowledge* resource.

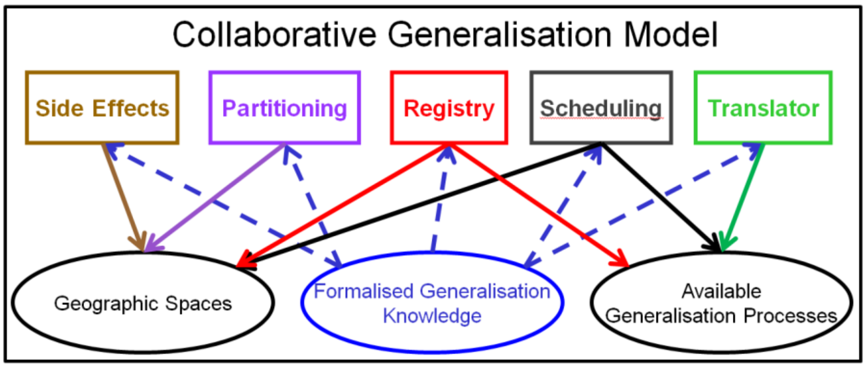

Collaborative Generalisation Model
Side Effects
Partitioning
Registry
Scheduling
Translator
Geographic Spaces
Formalised Generalisation Knowledge
Available Generalisation Processes

Fig. 2. The main Components (rectangles) and Resources (ellipses) of a Collaborative Generalisation Model and how the components act on the resources (plain arrows). The Formalised Generalisation Knowledge is used by all five components. The Side Effects and the Partitioning components act on Spaces while the Registry and the Iterating component act on both Spaces and Processes and the Translator only acts on Processes.

The geographic spaces being identified, we define a sequence of collaborative generalisation as a list of pairs (geographic space, generalisation process) interrupted by side effect processes. For instance a collaborative sequence could be: (Urban space 1, Process 1), (Urban space 2, Process 1), (Rural space 1, Process 2), side effects correction in Rural space 1 neighbourhood, (Mountain space 1, Process 3)...

The *registry* component aims at matching the pairs as yellow pages answering the question: what is the process to generalise this space? The registry records the services that the *available generalisation processes* are able to provide. Then, when a geographic space requests for generalisation, the registry component answers with a list of relevant processes. The registry mechanism is detailed more in section 4.4. The registry component clearly requires a description of the generalisation processes capabilities and needs to have access to user specifications to decide the application relevance of a process, both included in the *formalised generalisation knowledge* resource. The formal description of the generalisation processes capabilities is detailed in section 3.6.

The *scheduling* component chains the pairings of spaces and processes in an optimal sequence. It decides at each step which space has to query the registry for generalisation and evaluates the generalisation results. To iteratively choose the next space to be generalised, the scheduling component requires both user specification and general knowledge on the major steps of generalisation. The sequence is not linear but optimised by a trial and error strategy guided by general knowledge and online evaluation.

The *side effects* component relies on the observation of the neighbourhood of the spaces generalised by the *scheduling* component. The component monitors the potential side effects by triggering a deformation process as [11] that reduces conflicts without undoing the previous generalisations. The component requires to know user specifications in order to maintain the ones that are altered by side effects.

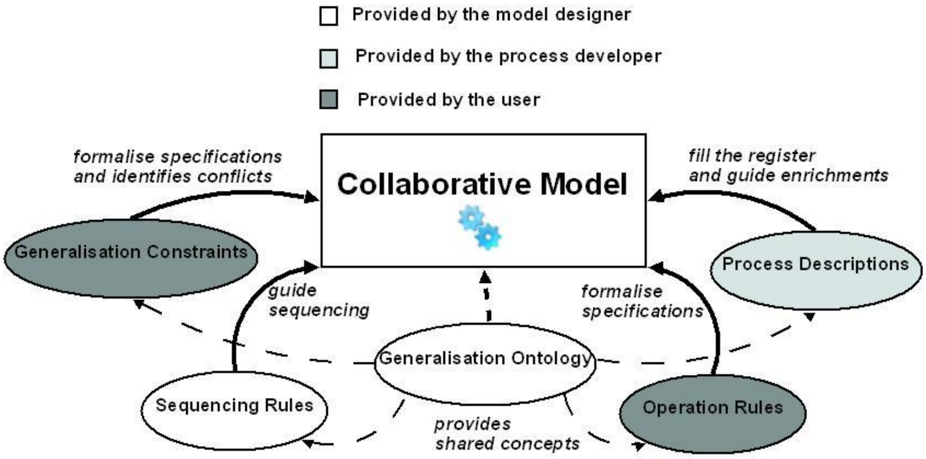
The *translator* component parameterises the available processes according to the user specifications whatever the process parameterisation system is. The translator component is detailed in sections 4.1 to 4.3.

Finally, the *formalised generalisation knowledge* resource gathers user specifications, generalisation processes descriptions and general knowledge on the scheduling steps of generalisation. We developed a Collaborative Generalisation model that relies on the components and resources described in this section, the *CollaGen* model (for Collaborative Generalisation). This paper focuses on the *formalised generalisation knowledge* resource modelling in *CollaGen*, presented in the next section. The interactions between the formalised knowledge and the translator and registry components are described in the fourth section.

3 Formalisation of Generalisation Knowledge

3.1 Organisation of the Formalised Generalisation Knowledge

In order to provide knowledge to the CollaGen model, user specifications and knowledge on cartographic generalisation are formalised in a machine interpretable way. User specifications cover here both the user requirements for the generalised map and the cartographic rules for map legibility. The formalised knowledge required for collaborative generalisation can be divided in five parts: generalisation domain ontology, generalisation constraints set, operation rules set, sequencing rules set and process descriptions.



□ Provided by the model designer
□ Provided by the process developer
■ Provided by the user
Collaborative Model
formalise specifications and identifies conflicts
fill the register and guide enrichments
Generalisation Constraints
Process Descriptions
guide sequencing
formalise specifications
Generalisation Ontology
Sequencing Rules
Operation Rules
provides shared concepts

Fig. 3. A diagram of the 5 parts of formalised knowledge and their use in the collaborative model. The dashed arrows show that the Ontology provides shared concepts to every part.

Fig. 3 shows how this formalised knowledge is organised to feed the collaborative process. The formalised knowledge is generated by three actors of the collaborative model that correspond to three times in the model life: the model designer that implements the five components and designs the generalisation ontology and the sequencing rules; the process developer that makes generalisation processes available and describe them, enriching potentially the ontology; the user that aims at generalising his data and then translates his specifications into generalisation constraints and operation rules.

The five following sections describe in detail the formalisation model of each piece of formalised knowledge and explain how the models are instantiated.

3.2 A Generalisation Domain Ontology

Automatic cartographic generalisation requires as input data an adapted data schema [12]. The adapted data schema is the initial schema of the geographic database used to produce the generalised map, enriched with implicit concepts made explicit in the data to allow the automatic process. The implicit concepts useful for automatic generalisation can be of different kinds: *meso* concepts [3] like “group of building”, “city” or “highway interchange”; *procedural* concepts that are necessary for the use of a particular process like the “fields” for a GAEL process [11], “dead ends” for a road selection process or “small compacts” for a CartACom process [3]; explicit *geographic relations* like the proximities between objects or the accessibility of a facility by a road. The generalisation constraints that mostly formalise the user specifications may concern the concepts and data added in the adapted schema.

We define a generalisation domain ontology as a domain ontology concerning the automatic generalisation process. The generalisation domain ontology should be made of:

- The concepts that can be present in an adapted data schema (meso, procedural concepts and geographic relations plus topographic concepts).
- The known relevant geographic spaces.
- The properties that may be constrained by user specifications.
- Generalisation operator taxonomy.
- A taxonomy of the generalisation processes available on the platform.

The geographic properties of concepts that are likely to be constrained by user specifications are included in the ontology as ontology properties. For instance, “area”, “granularity” or “absolute_position” are some of the properties defined on the “building” concept while “sinuosity”, “length” and “coalescence” are some “road” properties because constraints are often defined on these properties. Properties are also defined on the geographic relations: the “proximity” relation as a “minimum distance” property. The modelling of properties as results of spatial analysis measurements is advanced. Defining that shape should be measured by a mix of “compactness”, “concavity” and “elongation” properties is not possible yet. Describing more in detail the properties as in [13] would help to make a direct link between the atomic properties and the spatial analysis methods to measure the atomic properties and more abstract ones. Associations related to the adapted schema are also included in the ontology (e.g. the association "a *meso_entity* is composed of *geographic_entities*"). Some restrictions are defined on the associations. For example, the meso composition association can be restricted for building groups to only buildings and roads.

Our implementation of the generalisation domain ontology, in OWL 2, is built upon a topographic database concept taxonomy that was originally created in OWL by an automatic natural language process [14], manually enriched by the properties, associations and new concepts necessary to produce the generalisation domain ontology. The concepts possibly present in the adapted schema where classified using the national mapping experience of the laboratory. The generalisation operator taxonomy chosen as the most relevant for this ontology is extracted from [15] Then, the well-known meso, procedural and relations were added to the ontology with specific associations like a *meso_entity* “is composed of” *geographic_entities*.

The generalisation domain ontology is used as the support of generalisation knowledge sharing and integration, which is one of the applications of ontology [16].

3.3 Formalisation of Generalisation Constraints

In the first years of cartographic generalisation research, constraints have quickly been considered as the best way to formalise the map specifications [17]. Indeed, constraints, like “inter-distance between buildings must be at least 0.1 map mm”, are a convenient way to express the legibility conditions of a map. Generalisation constraints classifications were also suggested [3, 18]. Several research or production projects have proposed models to capture the user specifications in the form of generalisation constraints using table templates or OCL expressions [18, 19] while commercial software like Clarity™ (1Spatial) or Axpand® (Axes Systems) propose ad-hoc constraints expression models.

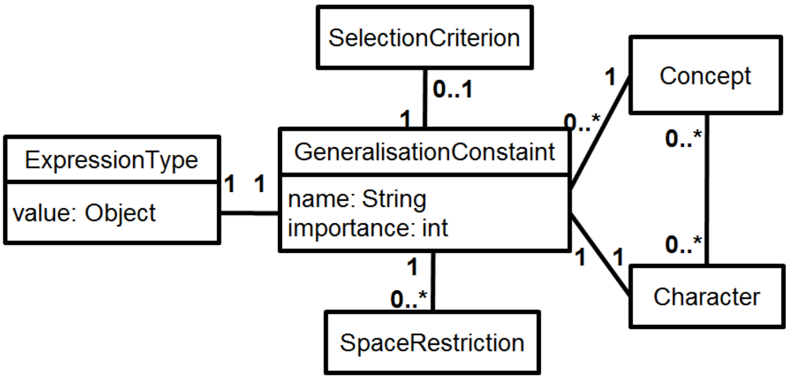


Fig. 4. UML class diagram of the Generalisation Constraints formal model. A constraint concerns a concept and one of its characters from the ontology and has an expression type, a selection criterion and a space restriction.

We developed a model to express the different user and map specifications as constraints that rely on the referred models and classifications (Fig. 4). Four types of constraints are defined from the classification of [18]: *micro* constraints (constraints on single objects), *meso* constraints (constraints on group of objects or patterns) and relational constraints (constraints on the geographic relation between two objects) and the macro constraints (constraints on the population of all objects of a kind). Only the last type is not present in the classification of [3]. Our contribution is the rest of the formal model described in Fig. 4. The model is described using the two following constraints examples:

- C1: "Buildings' area must be over 0.2 map mm^2 in urban areas".
- C2: "Very concave buildings should maintain initial concavity with 10% margin"

The major properties of a constraint are its *name*, and the *concept* ("building" for C1 and C2) and *character* constrained ("area" for C1 and "concavity" for C2) from the ontology. Then, generalisation constraints are characterised by an *expression type*, a *selection criterion* and a *space restriction*. The *expression type* is an object that holds both the kind of expression of the constraint and the threshold values. For example, the “threshold” type of expression means that the constraint is like: “concept.character < value”. Thus C1 has a "threshold" expression type with ">" as operator, "0.2" as value and "map mm^2" as unit. C2 has a "margin" expression type with a "10%" value. Five kind of *expression types* have been defined. The *selection criterion* is a query that selects one part of the objects of the constrained concept. For instance, C2 has a *selection criterion* that queries only very concave buildings (a threshold has to be given to translate "very" in understandable concavity value). The selection criterion can be seen has an implementation of the OGC Filter standard [20] for constraints. The *space restriction* is a set of geographic spaces from the ontology where the constraint is only applied. When the set is empty, the constraint concerns every part of space. Only C1 has a space restriction as the constraint is only valid in urban spaces.

A Graphical User Interface (GUI) form has been developed to help the user capture the constraints and implement the formal model. 70 constraints have been captured, extracted from French NMA experience.

3.4 Formalisation of Operation Rules

Although generalisation constraints may express most of user specifications, some part of the specifications cannot be appropriately expressed by generalisation constraints. Indeed, some of the constraints extracted from the EuroSDR test [18], particularly the ones advising or forbidding actions to apply ("...buildings should be aggregated"), are clearly rules that were forced to fit in the constraints template. So, we consider that it is simpler for a user to express them as Operation Rules. The rules are modelled following equation (1):

$$Premise_1 \wedge Premise_2 \wedge ... \Rightarrow Conclusion \tag{1}$$

Conclusions are generalisation operations from the ontology that are advised or not (e.g. “*Roundabout diameter < 100 m* implies *Collapse to point*”). A premise is a simple condition expressed with a threshold on a concept property, as in "threshold typed" constraints introduced in 3.3. Operation Rules can be seen as a convenient vector for modelling systematic operations as in the above roundabout rule. Operation rules are also a way to guide generalisation processes in their actions: the rule “buildings should not be aggregated in urban spaces” helps to parameterise the generalisation process that will be chosen to generalise urban spaces.

3.5 Formalisation of Sequencing Rules

The CollaGen model allows the expression of "sequencing rules" that represent general knowledge in automated cartographic generalisation and provide general guidelines to the scheduling component. The sequencing rules correspond to the *Global Master Plan* described in [21]. The Global Master Plan described how the main steps of generalisation are chained. For instance, the well-known rule “Network selection must be carried out before cartographic generalisation” can be expressed and processed thanks to sequencing rules.

As operation rules, the sequencing rules are modelled using premises and a conclusion. Fig. 5 shows the model of sequencing premises and conclusions. Premises refer to a particular place in the sequence of generalisation processes: “after network selection” or “when each part of the space has been processed once at least” are instances of particular places in the sequence. Conclusions can be either a geographic space (“Urban spaces should be processed first in cartographic generalisation”) or a process (“Geometry Collapses should be processed first”) from the ontology.

The implemented sequencing rules allow to sequence the generalisation process in four main steps: the geometry type changes (e.g. collapse of roundabouts to points), the selection (elimination of useless objects), the cartographic generalisation and the graphic generalisation [4] (correction of remaining legibility conflicts).

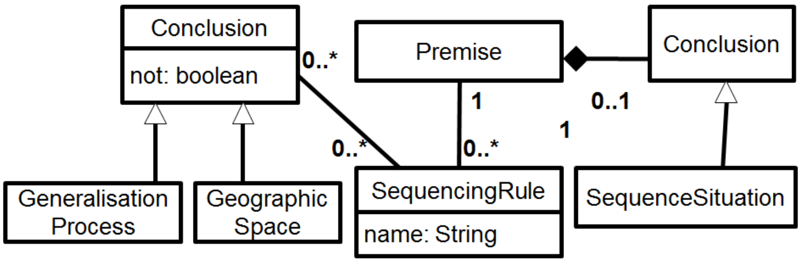


Fig. 5. The UML data schema of the Sequencing Rules model. A premise is a situation in the processes sequence and the conclusion is a generalisation process or a geographic space.

3.6 Formal Description of Generalisation Processes

As an analogy to web service composition, the composition of generalisation processes requires the description of their capabilities and requirements. The relevance domain of the different generalisation processes has to be formalised to know where they can be applied. For instance, we should be able to say that the CartACom process [3] is relevant on rural spaces or low density spaces and that the Elastic Beams [9] are relevant on flexibility graphs [22] (conflicting sub-graphs of the road network adapted to the Beams). We should also formalise which constraints can be handled by a process in order to know if it is adapted to particular situation. Regarding web service composition, the description of the service capabilities can be formalised by pre-conditions and post-conditions [23]. The pre-conditions correspond to the conditions the input data have to meet to be properly processed. The post-conditions describe the expected data modifications caused by the process. In the CollaGen model, this model is followed to describe the capabilities of generalisation processes in our collaborative model where pre-conditions are the relevant spaces for application and the post-conditions are the a priori handled constraints and rules (Fig. 6). Pre-conditions refer to spaces described in the ontology and post-conditions refer to constraints and rules present in the sets of constraints and rules defined by the user.

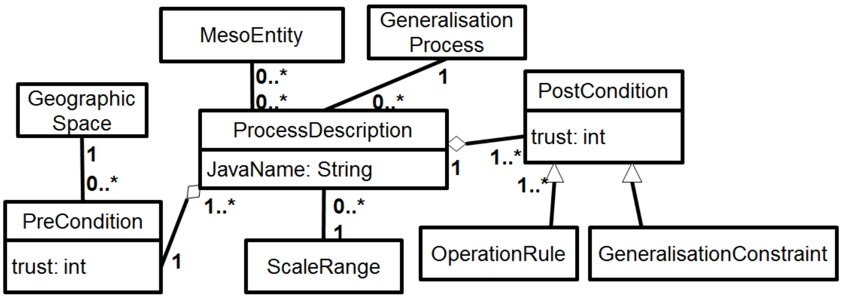


Fig. 6. UML class diagram of the generalisation process description for interoperability between processes. The pre-conditions are the spaces where the process is applicable and the post-conditions are the rules and constraints a priori satisfied after process execution.

To go further in the process description details, some properties are associated to the processes among which the generalisation method the process is an instance of (e.g. "AGENT specialised for urban generalisation" is an instance of "AGENT model"). It enables to link this process to the Sequencing Rules. The name of the programming component that allows to execute the process, is mentioned ("nameJava" attribute of class ProcessDescription in Fig. 6), which is a way of distinguishing *function* and *component* [24]. Added to that, the *scale range* class allows to define for the process the initial and final scales for an appropriate use of the process (e.g. the urban specialised AGENT process is appropriate for 1:10k to 1:50k). The limit scale ranges are also included in the class. Moreover, the required data enrichments to run the process (e.g. “dead-ends”, “road partition” or “building alignments”) are described in terms of meso or procedural ontology concepts that are expected to be added in the data. If the process is chosen by the *registry* component to generalise a given geographic space, the first step is then to process the enrichments on the space. Finally, the *trust* attribute on both *PreCondition* and *PostCondition* classes (Fig. 6) is an a priori evaluation of the relevance of each condition, provided by the process provider.

A GUI helps the process provider to fill the description that is automatically translated into the CollaGen description model. Eight generalisation processes available on our research platform are described including AGENT [3], CartACom [3], least squares [4], GAEL [11], elastic beams [9] and a road geometry collapse [5].

4 Processing Generalisations from Formal Knowledge

This section describes how the formal knowledge is used in the model by the *translator* and *registry* component. Section 4.1 deals with the need for matching the data schema to the ontology. Section 4.2 shows how the use of translator functions allows to trigger interoperable generalisations. Section 4.3 explains how, for a given geographic space, the relevant generalisation process is chosen. Some automatically triggered generalisation results illustrate the CollaGen model in section 4.4.

4.1 Matching Data Schema to the Ontology

Linking information resources (a geo-database schema here) to an ontology is made through a process called *annotation* [25]. In the CollaGen translator component, we used the annotation method called *registration mapping* that is a separate source containing the matching between schema elements and ontology concepts [25]. In the registration mapping, the useful ontology concepts, properties and associations are mapped to the equivalent in the data schema. For instance, the concept "road" is mapped to the class "BD_TOPO_Road_Section". We define the useful concepts as the ones that are actually used in the collaborative process (referred to in the constraints, operation rules and process descriptions).

Making the registration mapping automatically would require natural language processes that are not priority of this research so we opted for an interactive method.

For instance, a test case with one process, "urban AGENT" that requires the enrichment with building groups and three constraints on building minimal area, minimal granularity and inter distance, requires several mappings: first the ontology concepts "building" and "building group" have to be mapped to classes of the data schema; then the properties "area", granularity" on "building" and "building inter distance" on "building group" have to be mapped to the attributes of the data schema.

4.2 Translating Knowledge into Process Parameters

Once the objects of the database are matched to the ontology thanks to the registration mapping, the link between the objects and the constraints related can be made and so generalisation can be triggered. The generalisation processes first need to be parameterised according to the expressions and values held by the constraints and the rules captured by the user. As generalisation processes are very complex, they often require a big set of parameters and proper initialisations (e.g. defining constraints for AGENT, equations for the Least squares), giving importance to this translation step. We consider a process parameterised when all required parameters and initialisations have been set up. Thus, registering a generalisation process to the CollaGen model also requires providing a translator component that is able to read the constraints and translate them into the process parameters. A translator function of the component can be considered as a simple programming interface that enables the publishing of the process as a service, which is a key point of geo-processing interoperability [24].

Each generalisation process is provided with its standardised translator function (Equation 2). The body of the translator function consists in searching, for each parameter, for a constraint or rule in the sets that correspond to the parameter and in getting the value held in the constraint as the parameter.

$$\text{parameterised process} = \mathrm{f}(p, C, R, rm). \qquad (\mathbf{2})$$

Where *p* is the process to be parameterised, *C* is the constraints set, *R* is the operation rules set and *rm* is the registration mapping.

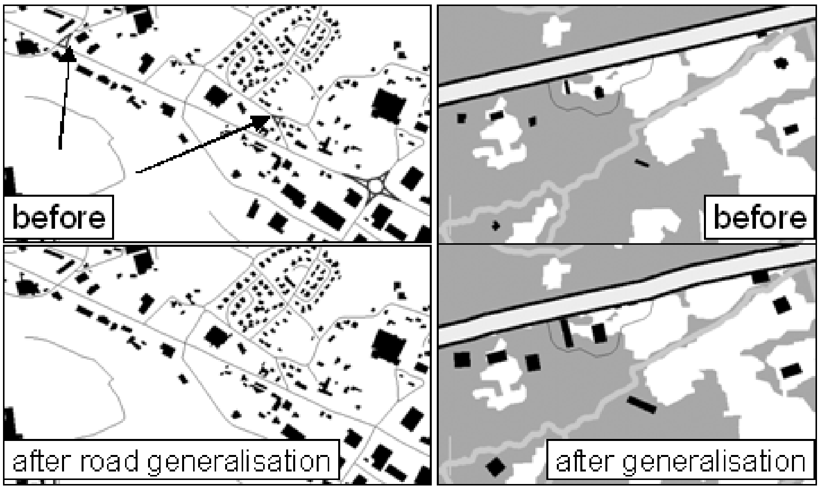


Fig. 7. Generalisation results from two different processes parameterised automatically by the formal knowledge and the translators. On the left, a road geometry collapse process parameterised with the translation of two rules concerning roundabouts and branching crossroads (highlighted with arrows). On the right, a least squares process parameterised with constraints on proximity **b**etween roads and buildings and on building size and granularity.

We developed the translator functions for the 8 generalisation processes available on our platform. For instance, the road geometry collapse process has simply real threshold parameters while the CartACom process is parameterised with constraints and the least squares process with equation systems. Fig. 7 shows two generalisation results from these three processes obtained with automatic trigger and parameterisation from the formal knowledge we captured to test our model, and the translator. A third result obtained with CartACom process is presented in Fig. 8.

4.3 Choosing the Best Process to Generalise a Geographic Space

As mentioned in section 2, the generalisation process descriptions are stored in a yellow pages registry that can be consulted to find the best process to generalise a geographic space designated by the *scheduling* component. The CollaGen implementation of the registry responds to a request with a list of relevant processes in relevance order. As in web search engines, the registry response is divided in two steps, the filter step that selects only the relevant services and the ordering step that orders the filter response in terms of relevance. In CollaGen, the filter step questions the pre-conditions of the descriptions (i.e. the geographic spaces a priori accepted as possible input for the process), and keeps the processes whose pre-conditions correspond to the space concerned by the request. For instance, if only the AGENT process and the least squares process have "urban space" in their pre-conditions, an urban space requesting a generalisation will only get these two answers. As a first approximation, the ordering step of the request is made in two times. A first ordering is made according to the "trust" value (integer between 1 and 5) of the pre-condition. Then, pre-conditions with the same trust value are ordered according to the post-conditions, that are the constraints and rules a priori satisfied. The more the post-conditions match the actual constraints conflicts, the best the process is rated.

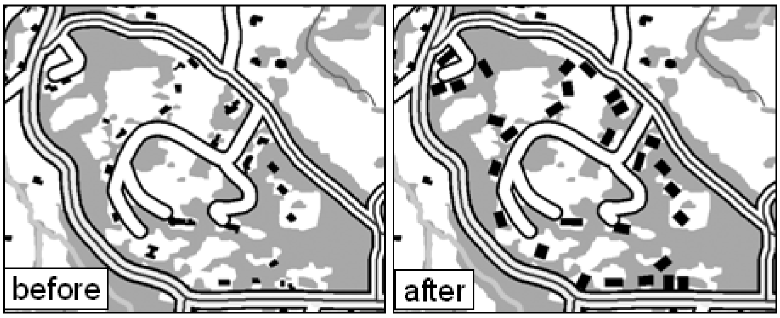


Fig. 8. A Rural space (built by the partitioning component) generalised by the CartACom process according to the request to the registry of generalisation processes.

Fig. 8 illustrates the choice of the best process to generalise a "Rural" geographic space. Three of the eight available processes have a pre-condition about rural spaces: "CartACom" (trust value of 4), "Urban AGENT" (trust value of 2) and "least squares" (trust value of 2). "CartACom" is put on top of the list and tried first. The ordering of the two remaining ones is done comparing the conflicts in the rural space to the post-conditions. For instance, the preservation constraint "preserve parallelism between roads and buildings" causes conflicts that should be dealt by the post-conditions. Finally, the “least squares” is advised first for a better preservation of the parallelism constraint. Anyway, as the generalisation with the CartACom process is evaluated as satisfying, the following propositions in the list are not considered.

4.4 Some Results With Several Processes

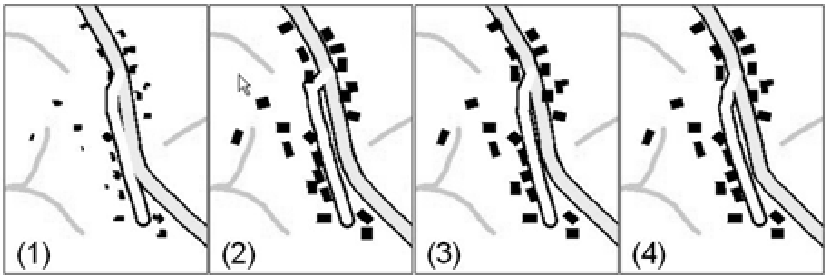


Fig. 9. (1) a situation before generalisation. (2) the situation generalised with an AGENT based process then a Least Squares process: some conflicts remain. (3) the situation generalised with a CartACom process then a Beams process: some side effects are created by the beams. (4) the situation generalised with CartACom then Least Squares: it is correctly generalised.

Although the CollaGen model is not fully implemented some results can be presented. Fig. 12 shows several processes parameterised by the translator that are executed on the same situation. The four processes used in this example are fully interoperable within CollaGen and we can see that the third sequence gives the best results. It is hopefully the first one proposed by the registry regarding the rural space the situation is in: the registry proposes CartACom with the rural space in firstly generalised then proposes the Least Squares as the best process for final graphic generalisation [4].

5 Conclusions and Further Work

The paper introduced and defined the collaborative generalisation approach. In such an approach the initial data is partitioned in different geographic spaces (cities, countryside, mountains, etc.) that are generalised by the more appropriate of the available automatic processes while side effects between spaces are controlled. We presented an important aspect of the CollaGen model (our implementation of collaborative generalisation): to enable the interoperability of the processes and the homogeneity of the generalisation, cartographic knowledge and user specifications are formalised in constraints and operation rules sets, sequencing rules and process descriptions, all based on generalisation domain ontology. Once the initial data is annotated with the ontology, translator components allow parameterising the processes and the processes can be chosen and triggered on a given geographic space.

To go further, some classical generalisation constraints could be integrated in the ontology to ease the capture of specifications by the user. But before, two topics have to be tackled more deeply to make the CollaGen model operational. First, the management of the side effects has to be clarified: when do we exactly need to trigger the correction and how do we observe the related conflicts? Then, the scheduling component implementation has to be finalised.

References

1. Touya, G.: First thoughts for the orchestration of generalisation methods on heterogeneous landscapes. In: 11th ICA Workshop on generalisation and multiple representation, Montpellier, France (2008) http://aci.ign.fr/montpellier2008/papers/01_Touya.pdf
2. McMaster, R. B., Shea, K. S.: Cartographic generalization in digital environment: A framework for implementation in a gis. In: GIS/LIS'88, 240--249 (1988)
3. Ruas, A., Duchêne, C.: A Prototype Generalisation System Based on the Multi-Agent System Paradigm. In: Mackaness, W., Ruas, A., Sarjakoski, T. (eds.) The Generalisation of Geographic Information: Models and Applications. Elsevier. 269--284 (2007)
4. Harrie, L., Sarjakoski, T.: Simultaneous Graphic Generalization of Vector Data Sets. Geoinformatica 6(3), 233--261 (2002)
5. Touya, G.: A Road network Selection Process based on Data Enrichment and Structure Detection. Transactions In GIS, in press (2010)
6. Weiss, G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press (2000)

7. Regnauld, N.: Evolving from automating existing map production systems to producing maps on demand automatically. In: 10th ICA Workshop on Generalisation and Multiple Representation (2007) http://aci.ign.fr/BDpubli/moscow2007/Regnauld-ICAWorkshop.pdf
8. Duchêne, C., Gaffuri, J.: Combining Three Multi-agent Based Generalisation Models: AGENT, CartACom and GAEL. In: Ruas, A., Gold, C. (eds.) Headway in Spatial Data Handling, 13th International Symposium on Spatial Data Handling. LNG&C, pp. 277--296. Springer, Heidelberg (2008)
9. Bader, M., Barrault, M.: Cartographic Displacement in Generalization: Introducing Elastic Beams. In: 4th ICA workshop on progress in automated map generalisation, Beijing (2001) http://aci.ign.fr/BDpubli/pbeijing2001/papers/bader_barraultv1.pdf
10. Chaudhry, O. Z., Mackaness, W. A.: Creating mountains out of mole hills: Automatic identification of hills and ranges using morphometric analysis. Transactions in GIS 12 (5), 567—589 (2008)
11. Gaffuri, J.: Three reuse example of a generic deformation model in map generalisation. In: 24th International Cartographic Conference, Santiago, Chile (2009)
12. Ruas, A.: Automating the generalisation of geographical data: the age of maturity? In: 20th International Cartographic Conference, Beijing, China, 1943--1953 (2001)
13. Lüscher, P., Weibel, R., Mackaness, W. A.: Where is the terraced house? On the use of ontologies for recognition of urban concepts in cartographic databases. In: Ruas, A., Gold, C. (eds.), Headway in Spatial Data Handling. 449--466 (2008)
14. Abadie, N., Gesbert, N., Mustière, S. : Création d'une ontologie à partir des spécifications textuelles pour l'intégration des bases de données géographiques. In : 17èmes journées Ingénierie des Connaissances, Nantes (2006)
15. Mustière, S.: Apprentissage supervisé pour la généralisation cartographique. PhD thesis, Université Pierre et Marie Curie (2001)
16. Horrocks, I.: Ontologies and the semantic web. Commun. ACM 51 (12), 58--67 (2008)
17. Beard, K.: Constraints on rule formation. In: Buttenfield, B., McMaster, R. (eds.) Map Generalization, 121--135, Longman (1991)
18. Burghardt, D., Schmid, S., Stöter, J.: Investigations on cartographic constraint formalisation. In: 10th ICA Workshop on Generalisation and Multiple Representation, Moscow (2007) http://aci.ign.fr/BDpubli/moscow2007/Burghardt-ICAWorkshop.pdf
19. Stoter, J.E., Morales, J.M., Lemmens, R.L.G., Meijers, B.M., van Oosterom, P.J.M., Quak, C.W., Uitermark, H.T., van den Brink, L.: A Data Model for Multi-scale Topographical Data. In Ruas, A., Gold, C. (eds.), Headway in Spatial Data Handling, 233--254 (2008)
20. Filter Encoding Implementation Specification, http://www.opengeospatial.org/standards
21. Ruas, A., Plazanet, C.: Strategies for Automated Generalization. In: 7th International Symposium on Spatial Data Handling, 319--336, Delft, Netherlands (1996)
22. Lemarié, C.: Generalisation process for top100: research in generalisation brought to fruition. In: 5th ICA Workshop on progress in automated map generalisation (2003)
23. Lutz, M.: Ontology-Based Descriptions for Semantic Discovery and Composition of Geoprocessing Services. GeoInformatica 11(1), 1--36 (2007)
24. Bucher, B., Jolivet, L.: Acquiring service oriented descriptions of GI processing software from experts. In: 11th AGILE International Conference, Girona, Spain (2008)
25. Lemmens, R.: Lost and found, the importance of modelling map content semantically. In: Peterson, M. (eds), International Perspectives on Maps and the Internet, 377--396 (2008)