



Integrating structured descriptions of processes in geographical metadata

Bénédicte Bucher

► To cite this version:

Bénédicte Bucher. Integrating structured descriptions of processes in geographical metadata. Developments in Spatial Data Handling, Springer-Verlag, pp.629-639, 2005, 10.1007/3-540-26772-7_47 . hal-02435275

HAL Id: hal-02435275

<https://hal.science/hal-02435275>

Submitted on 10 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Integrating structured descriptions of processes in geographical metadata

Bénédicte Bucher

Laboratoire COGIT, Institut Géographique National, 2 avenue Pasteur, 94 165 St Mandé Cedex,
France, benedicte.bucher@ign.fr

Keywords : metadata, process, task

Integrating structured descriptions of processes in geographical metadata

Abstract.

This paper extends upon a category of information, processes description, and its relevance in metadata about geographic information. Metadata bases about processes are needed because processes are themselves resources to manage. Besides, specific processes participate in the description of other types of resources like data sets. Still, current metadata models lack structured containers for this type of information. We propose a model to build structured descriptions of processes to be integrated in metadata bases about processes themselves or about data sets.

Introduction

During the past five years, significant progresses have been made in modelling metadata to enhance the management of geographical data. The release of the ISO19115 international standard was an important milestone. Metadata models initially aimed at resolving data transfer issues. Their application is now to support data exchange and cataloguing. The very scope of these models has widened from describing geographical datasets to describing aggregates of data sets, i.e. specifications of representation, and services. This paper extends upon the necessity of enriching geographic information metadata with structured descriptions of processes.

The first section of this paper details the need for this category of knowledge in metadata about geographical information. The next section is a brief state of the art of existing models to describe processes, including a prior work of the author in this domain. The section after describes our current approach that focuses on describing data production and management tasks to enrich metadata bases within IGN.

The need for structured descriptions of processes in metadata

In this paper, studied processes are manipulations of geographical data to meet an objective. The need for structured descriptions of processes in metadata is twofold.

Processes management metadata

Processes themselves are resources to manage so that a metadata model is needed to describe them.

Managing a process usually means to track its running or to assist operators in performing it. For instance, the process of acquiring metadata about a data set could be designed as a process distributed between several people participating in the production of the data set. In this case, metadata associated to the data production process would be of the following forms :

- ~ flags warning to perform a data metadata acquisition event associated to certain events in the data production process,
- ~ guidelines to perform these metadata acquisition events.

Managing a process can also mean cataloguing it. The famous three stages of data cataloguing proposed by the Global Spatial Data Infrastructure Technical Committee (GSDI 00), discovery, exploration and exploitation, can be translated as follows for processes cataloguing :

- ~ discovery : What processes exist?
- ~ exploration : Which processes are close to the process I need?
- ~ exploitation : Can I use this process? Can I adapt it to my context? Or can I define a new process through reusing existing processes patterns?

Processes that typically need cataloguing are repair processes. Repairing a given error consists in performing an existing repair process when the error is already referenced. When it is not, it may consist in designing a new repair process possibly based on catalogued diagnosis and repair processes.

Data sets description metadata

Modelling processes in metadata is also needed to manage other types of resources in the description of which processes appear. Indeed, descriptions of processes are already present in the ISO19115 model where the lineage entity is composed of sources and processes. This specific category of metadata, lineage, is essential to IGN sale engineers. They use it as a major source to assess the content and quality of a data product or data set. They don't find so far this information in metadata bases but rather by contacting production engineers.

The ISO19115 structure for processes elements in the lineage entity is free text. Obviously, this information calls for a richer structure.

A need for a model of processes types and instances

To conclude this section, let us summarize what structures are needed to describe processes in metadata, as illustrated on **Fig. 1**. Integration of structured descriptions of processes in metadata about geographic information.

- ~ . A model is needed to describe :
- ~ processes types, like data matching,
- ~ processes instances, like consistency checking for two specific data sets.

These structures are needed to build process management metadata.

Descriptions of specific types of process constitute a metadata base about these types of processes. It could entail information like the name of the process, its generic signature, its decomposition. Metadata bases about specific instances of a process can then be obtained through specifying elements in the description of the corresponding type of process, like the name of the operator who was responsible for performing this process instance and some decisions he made during the process.

These structures are also needed to document the lineage metadata in data descriptions.

A type of process can be the production of a type of data sets. For instance the Scan25® product in IGN is a data set series associated to a type of production process. The description of this type of process is a lineage metadata for this specific aggregate.

The lineage metadata for a data set belonging to this aggregate is then the description of the specific process instance which output was this data set.

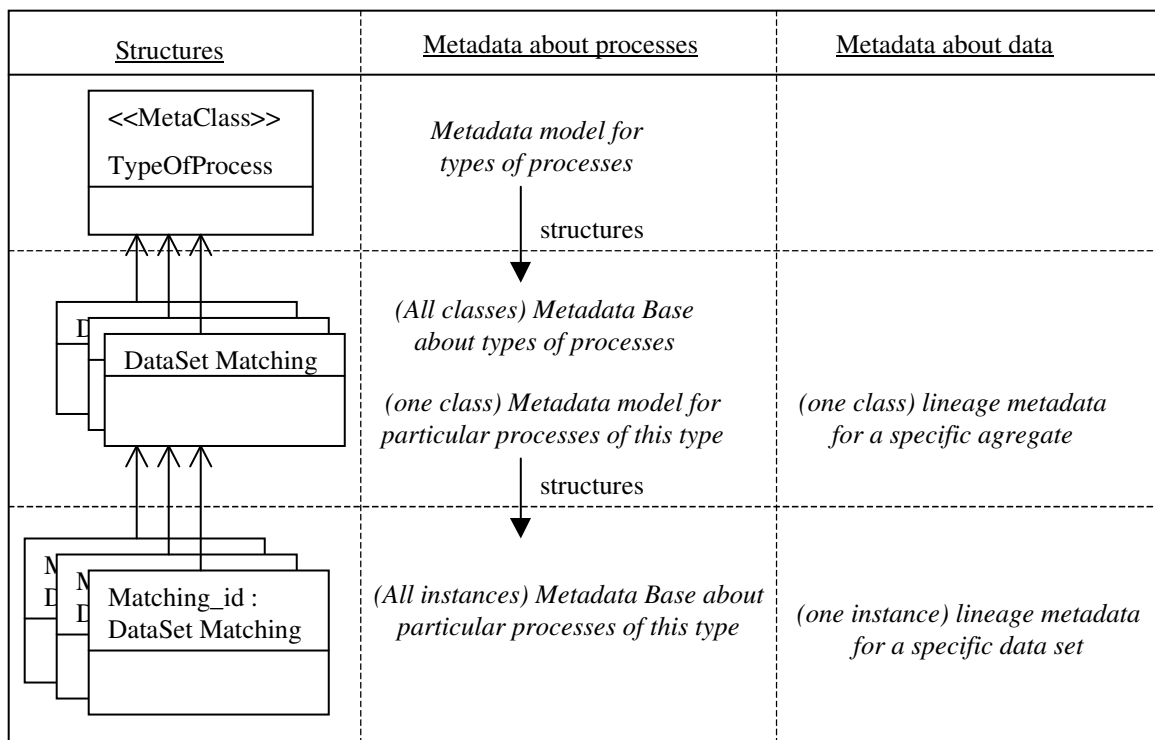


Fig. 1. Integration of structured descriptions of processes in metadata about geographic information.

Existing models to describe processes

Models to describe processes as such are to be found in the domain of business management. They integrate components like triggering events, decomposition and agents (Tozer 99).

In the context of the Web, there exist models to describe Web services. Web services realise specific categories of process. In this area, the most achieved approach is that of OWL-S (OWL-S 02). OWL is a W3C effort to propose languages to build ontologies on the Web. A section of OWL is specifically dedicated to Web Services. It proposes the following RDF-like model to describe services. A resource provides a service. A service presents a service profile, i.e. what the service does. A service is described by a service model, i.e. how it works. A service supports a service grounding, i.e. how to access it.

In the area of geographic information, there exist classifications of geographical Web services like that of OGC (OGC 02). A promising work is that of (Lemmens et al.03) who extend the OGC Web service taxonomy schema using DAML-S (former name of OWL-S).

All these models support the description of processes instances or of very specific process types like a Web Service is. But they don't provide much description for more generic process types, apart from classifications.

In the COGIT laboratory, the TAGE model has been proposed to enrich ISO19115 metadata with how-to-use knowledge (Bucher 03). This model describes application patterns like to locate an entity or to match data sets. These applications patterns can be specified to obtain a description of a geographic application that should yield the result the user expects. TAGE relies on the classical concepts of tasks and roles. A task is a family of problem and their solution. A task is a process type, generic or specific. It can be specified to a more specific process type or to a process instance. A process instance is a realisation of a task. The inputs and output of tasks are described through roles. A role is a variable with a name and a set of possible values. For instance, the output of the task « to locate an entity » is a role called « location ». The set of possible values for this role includes : a geometry in a spatial reference system, coordinates in a linear reference system, a place defined by a relationship with a geographic feature, a symbol on a map or route instructions. The role itself is task dependant whereas the elements describing its values or not task dependant. These elements are called "domain elements". TAGE has been embedded in the TAGINE application to assist users in specifying a task which realisation should meet their need.

Tasks and roles are relevant concepts in the description of generic processes. The TAGE model supports the description of tasks which decomposition varies depending on how they are specified, which is not supported by other decomposition models. Moreover, the use of roles and of specification rules embedded within the task allows for the consistent specification of the output depending on the specification of the input. This is far more precise than describing the signature of a process. Indeed, the effect of a process, as well as other properties of the process, may depend on the input data. The limitations of TAGE are the complexity of this model and the consequent difficulty of acquiring tasks to feed in the TAGINE database.

TAGE2 : a simplified TAGE model

Our current approach to describe processes consists in simplifying the TAGE model to obtain a model and a database that are more readable and easy to maintain.

A shift of objective

The corresponding application does not meet the same objective as TAGINE. Our ambitions have shifted from enhancing external users access to geographical information to enhancing internal users access to geographical information. This takes place in a new context in IGN where people are more aware of the importance of metadata. This context is described in (Bucher et al. 03). Technically, TAGINE aimed at supporting cooperative specification of a task. The new application only aims at : browsing tasks, editing a task and interactively specifying it, storing tasks and storing realisations of tasks.

The representation of tasks in TAGE2

In the initial TAGE model, tasks were to be modelled as instances of one class, similar to the ISO FC_FeatureType structure. In TAGE2, a task can be modelled as a class extending the class “Task” or as an instance of the class “TaskType”, as shown on **Fig. 2**. Double view on tasks : a specific task can be described as a class extending the class Task or as an instance of the class TaskType.. The same holds for domain elements with the classes “DomainElement” and “ElementType”. We use this because the java language does not support the representation of metaclass. Still metaclass like structures are needed as explained in section 0. A specific task should be both an instance of another class and a class itself.

Describing a task as an instance of the class TaskType, or of a subclass of it, is useful to describe numerous tasks, typically to build a metadata database about processes. Describing a task as a class is useful to focus on the model of the specific task. Instances of this task will be descriptions of processes instances. These needs were listed on Fig 1. The use of TAGE2 structure to meet these needs is summarized on Fig 3.

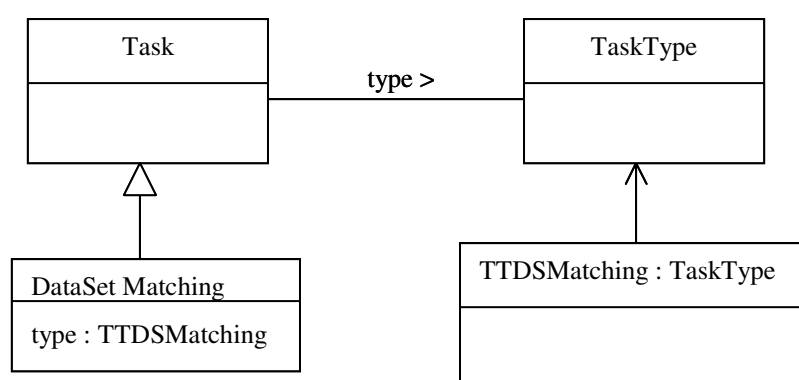
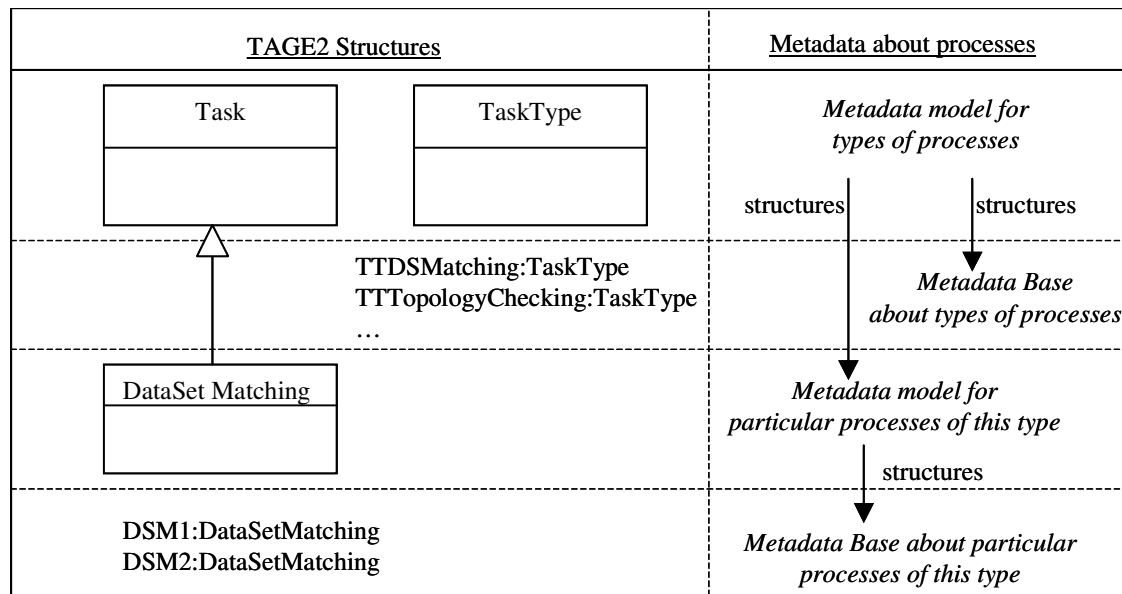


Fig. 2. Double view on tasks : a specific task can be described as a class extending the class Task or as an instance of the class TaskType.

Fig. 3. The use of TAGE2 structures to meet the needs expressed in the first section



A prototype

To build a prototype of TAGINE2, we have chosen to describe a specific data management task : data set matching. This task is used by the data producer to assess the quality of a data set, to update a data set or to build a multi-representation data set.

The domain elements for this task have been modelled according to the implementation of ISO standards in the laboratory plate form OXYGENE (Badard and Braun 2003). We introduced the concept of data set and related it to features implementing the ISO FT_Feature interface. A data set is also associated to unit of storage that can be XML files or Oracle tables or directories. It has a description that is a ISO19115 MD_DataDescription entity. These elements are illustrated on Fig 4. Other domain elements relevant to the data set matching task are Feature catalogues that are referred to in MD_Description entities.

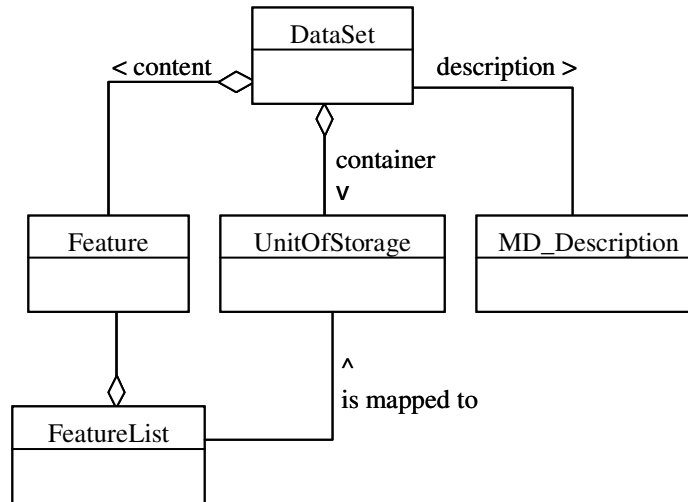


Fig. 4. Domain elements representing a data set.

The overall mechanism of the task is to find minimas of the distance between two representations of the same space, at the model level and at the data level. Its decomposition is the following.

The first subtask is to match schemas. This consists in matching groups of elements in both schemas that represent the same category of objects in reality. In this task, it is also important to identify relationships and attributes that don't vary with the representation and to mark those that are identifiers. This task should rely on stored correspondences between feature types in feature catalogues and on stored marking of non varying attributes and relationships in each feature catalogue. In the future, these elements will be integrated in the domain.

The next subtask is to rank the schemas correspondences after the evaluated easiness and quality of data matching for the corresponding features. If two classes are matched that bear a non varying identifier, then features corresponding to these classes should be matched first. If two classes are matched and bear numerous non varying relationships, the corresponding features should be matched early. Indeed, these first results will then be used to match features related to them by these relationships. There exist numerous rules like these ones that should be applied to build a tree of correspondences.

The last subtask is to match features. It is itself decomposed into three tasks : to select a feature in the reference data set –after specific criteria–, to restrict the compared data set to features that may be matched to the selection and to use geometric algorithms to assess the correspondence.

The Fig 5 shows task browsing supported by the prototype. A task is described by its roles, a brief description of its model, and a summarized description of its decomposition. This decomposition can be browsed more in detail in another window. The “detail” buttons next to each role open windows describing the possible values for the role.

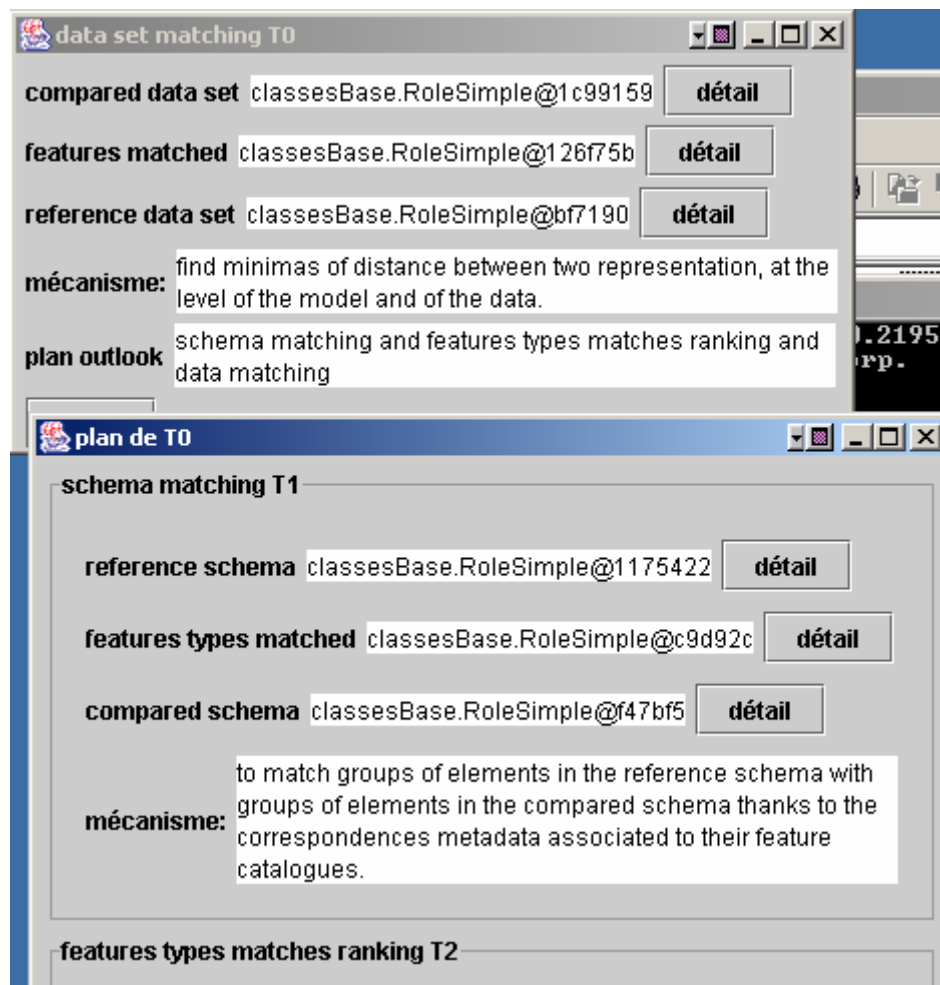


Fig. 5. Browsing of the DataSetMatching task in the prototype. The interface is in french but some elements (the tasks) have been translated into English for the purpose of this paper.

Perspectives

On-going work aims at providing a simple interface to edit tasks as well as domain elements.

The next step will consist in acquiring tasks and realisations of tasks.

A first method to acquire tasks consists in asking people to describe through the TAGINE2 interface a task they are familiar with, either by building a new instance of TaskType that will possibly reuse other instances, or by building a new subclass of Task that will possibly reuse other subclasses.

Another method we plan to experiment is the analysis of a set of log files corresponding to the same task to extract common patterns.

Acquiring realisations of a specific task will rely on the description of this task as a specific class. We will focus on supporting this acquisition in a distributed context, i.e. when it is performed by several people. Again this may be done through these people using the TAGINE2 interface or through the analysis of their log files.

We intend to focus on the acquisition of data production tasks and on the integration of these descriptions in the ISO19115 lineage entity.

Acknowledgements

The author wishes to thank Sandrine Balley, Sébastien Mustière and Arnaud Braun from the COGIT laboratory for their help in modelling the Data Set Matching task, its roles and decomposition, and the corresponding domain elements.

References

- (Badard and Braun 03) Thierry Badard, Arnaud Braun, OXYGENE : An Interoperable Platform Enabling the Deployment of Geographic Web Services, GISRUK conference, London, 2003
- (Bucher 2003) Bénédicte Bucher, Translating user needs for geographic information into metadata queries, 6th AGILE Conference, Lyon, 2003, pp.567-576.
- (Bucher et al. 03) Bénédicte Bucher, Didier Richard, Guy Flament, A metadata profile for a National Mapping Agency Enterprise Portal - PARTAGE, GISRUK Conference, Londres, 2003
- (GSDI 00) GSDI Technical Working Group, Developing Spatial Data Infrastructures : the SDI Cookbook, v1.0, Douglas Nebert (Ed), 2000.
- (Lemmens et al. 03) Rob Lemmens, Marian de Vries, Trias Aditya, Semantic extension of Geo Web service descriptions with ontology languages, 6th AGILE Conference, Lyon, 2003, pp.595-600
- (OGC 02) The OpenGIS Consortium, OpenGIS® OWS 1.2, 2002
- (OWL-S 03) The OWL Services coalition, OWL-S 1.0, Semantic Marks up for Web Services, 2003
- (Tozer 1999) Guy Tozer, Metadata management for information control and business success, Artech House, Boston, 1999