

# STRUCTURING AND ENRICHING METADATA TO ENABLE USERS' ACCESS TO GEOGRAPHIC INFORMATION RESSOURCES

Bénédicte Bucher  
COGIT – IGN  
2 av Pasteur  
94 165 St Mandé Cedex , France  
Fax : +33 1 43 98 81 71, [benedicte.bucher@ign.fr](mailto:benedicte.bucher@ign.fr)

**Abstract :** A crucial requirement for the effective exploitation of geographic information is users' access to the knowledge held in corresponding stored resources, i.e. their understanding of what can be derived from available geographic databases and software. At the core of it is the problem of how to share and reuse knowledge, a major current challenge of Artificial Intelligence (AI). The work presented here extends upon lessons learnt in AI. The proposed solution to the users access issue is an expert system that stores geographical knowledge derivation recipes, and that allows users to browse and customise these recipes. Specifically, we build a model representing the recipes components, i.e. geographic application patterns. This model structures metadata about geographic data and metadata about stored processes, and it describes the usage of these data and processes. A prototype of this system, coded in java, is presented in this paper.

## 1 INTRODUCTION

### 1.1 Context : access to stored geographical information

The building and maintenance of stored geographical information, be it data sets or processes, generate numerous industrial and research efforts. Good use of these resources is more and more crucial. Not only does the value of information mainly lies in its being useful to end users, but also the current paradigm of information and the development of numerical geographic information resources are such that "g business is everywhere" as puts it [Rhind 01].

To enhance users' access to geographic information, spatial data infrastructure are built at national levels, relying on metadata standards for the users to identify relevant data for their needs. Also warehouses are built to store geographic data and sometimes other related information. Yet access does not only mean *to know what resource exists, what its content is and where to retrieve it* but also : *to know how to use a resource, and to know how to use several resources together*. An issue impossible to circumvent, in this process, lies in a specificity of geographic information summed up hereafter.

The space we live in is perceived in many way and there is no 'natural' universal model of a geographic world. Its representation in stored information resources derives from modelling considerations, but also from acquisition constraints as well as paradigms of the resources producer. This is especially true when it comes to numerical resources. The users of this information have their own perception of the geographic world. Besides, an application domain often comes along with its own family of models and representations, that lend themselves to the reasoning performed in this domain. Users' access imply then to understand the model and representation used to build the resources and to relate them with the own model of the users. Understanding how to use geographic data is then pretty often a tricky issue for the non specialist. The semantic interoperability issue is rampant in geographic information [Bishr 97]. Moreover, because of the complexity of geographic data types and relationships, "the GIS design task [is] a process closer to the implementation than to a software engineering process" [Balaguer et al ].

### 1.2 A general underlying issue : to access and reuse stored information

#### 1.2.1 The issue

The problem enounced above is a particular case of users' access to stored information resources. This access can be seen as the process of a user browsing, in a focused way, the space of what he can obtain through manipulations of the stored information. AI techniques to enhance knowledge sharing and reuse provide a theoretical basis on which to build systems which can assist in this browsing process.

### 1.2.2 Lessons learnt from AI

As recalled in [Gomez and Benjamins 99], in the context of knowledge sharing and reuse, lessons learnt from AI are to describe the knowledge to be shared in terms of :

- essential characteristics of the elements of the domain, i.e. ontological terms,
- how elements of this domain can be used, i.e. problem-solving terms.

Software engineering techniques, like the Unified Modeling Language (UML), do not yet provide a model to integrate both types of terms. Such models are rather put forward by knowledge engineering techniques. In this area, the notion of “task”, i.e. the description of a problem and of how to solve it, is a promising technique to make a connection between domain knowledge, or ontological terms, and problem-solving methods [Chandrasekaran 98].

A task-based model is used to describe resources by giving examples of tasks that can be achieved with the resources. It intuitively corresponds to the following approach : to give the user an overview of possible use-goals, i.e. a “*why*” description, and to teach the user methods to handle the resources, i.e. a “*how*” description of the resources. It obviously necessitates to have a description of the resources themselves, i.e. a “*what*” description. All types of descriptions are needed by users of geographic digital resources.

There are several successful experiments at building such models to share knowledge. Ontoseek is a search engine for object-oriented software components relying on a functional description of such components [Guarino et al. 99]. The *Atelier Logiciel* is an expert system which describes pieces of image processing codes, in a “image-dedicated” vocabulary and using a task-based model. Users who are not experts in image processing and who need to process images in their work, e.g. in medicine or optical surveillance, use this system to specify an application combining pieces of image-processing codes [Ficet et al. 99].

## 1.3 Approach

The approach presented in this paper extends upon lessons learnt in AI to address the case of geographic information [Bucher 00]. We aim at enhancing users' access to geographic data and processes by supporting their browsing of applications that can be built upon these data and software. To support this, we build a knowledge-base browser which integrates ontological and problem-solving like metadata. The functionality, built above this metadata structure, allows users to browse a description of applications they can build from these data and processes.

In this paper, we introduce what we call geographic application patterns and the model used to build the knowledge base of the browser. We then describe the object-oriented representation of this model and the prototype built using this representation.

## 2 A MODEL OF GEOGRAPHIC APPLICATION PATTERNS

### 2.1 Modelling usage patterns

#### 2.1.1 Existing geographic information usage descriptions

There are attempts at building different types of geographic information descriptions. There are too numerous to be listed here but we give a brief overview of the variety of these approaches. At the perception level, Gibson introduced the concept of *affordance* to describe the environment : "I have described environment [...]. But I have also described what the environment affords animals, mentioning the terrain, shelters, water, fire, objects, tools, other animals, and human displays. How do we go from surfaces, is there information for the perception of what they afford. If so, to perceive them is to perceive what they afford"[Gibson 79]. Some approaches focus on describing specific application domains, e.g. Corona and Winter build an ontology of pedestrian navigation in order to evaluate how far spatial data sets are far from the concepts useful for pedestrian navigation applications [Corona and Winter 01]. Numerous authors have tried to list manipulations of geographic objects in spatial analysis, and GIS usage. In the context of interoperability and information reuse, some authors recommend to use

functional languages to specify or to describe interoperable components [Vckovski 98] [Kuhn and Frank 97]. Balaguer, Gordillo and Das Neves also claim that “the GIS community should record its design expertise in terms of *Design Patterns*“ in a reusable way for minimising the task of designing a GIS application [Balaguer et al. 97].

Our approach is to build a model integrating ontological and problem-solving knowledge at the metadata-level, interfacing end users and geographic data and software components. The patterns we want to store and reuse are what helps in answering the following questions. *Why?* : what presentation of information does the user want? *How?* : what underlying creation of information is needed? *What ?*: what stored data and processes should be retrieved ?

Ontological terms describing the data, the *what*, are mainly held in classical metadata. These classical metadata do not account much for the way the data should be used. Problem-solving knowledge, the *why* and *how*, need to be represented in a new type of metadata. All categories of knowledge are integrated in one model, exposed in the rest of the paper, as geographic application patterns.

### 2.1.2 The CommonKADS model of expertness

We build the knowledge model of our system after the expertness model proposed in the CommonKADS project [Schreiber et al. 99]. This model is structured in three categories shown fig. 1. The Task component models *why* knowledge, its method model *how* knowledge. Inferences, roles and transfer function components model *how* knowledge and the domain models the *what* knowledge.

Category	Construct	Description
Task knowledge	<i>Task</i>	A problem statement of what needs to be achieved; specifies also input and output
	<i>task method</i>	Specifies a way to achieve a task by decomposing it into subtasks, inferences and transfer functions; also defines a control regimen over the decomposition
Inference knowledge	<i>Inference</i>	A primitive reasoning function which achieves a basic problem-solving step
	<i>Role</i>	Input or output of an inference; signifies a place holder and an abstract name for domain objects
	<i>Transfer Function</i>	Used to denote a primitive function needed to that interact with the outside world
Domain knowledge	<i>domain schema</i>	A set of domain-type definition
	<i>concept</i>	A group of “things” with share features
	<i>Relation</i>	Describes a set of rules that relate “things” to each other
	<i>rule type</i>	Antecedent/consequent expressions
	<i>Knowledge base</i>	Set of domain-type instances

Fig. 1. : Constructs in CommonKADS knowledge model (from [Schreiber et al. 00])

CommonKADS offers components to explicit useful knowledge in a model, but it does not offer representation components to code this model. We have chosen the object-oriented language to represent our model, and the java language to code the first elements of our representation in a prototype. In the next section we present the main lines of this representation.

## 2.2 Objects to represent a KADS model of geographic information expertness

### 2.2.1 The Task concept

Two points of view are grouped in the single task concept :

- The declarative point of view makes up for the specification of the task. This consists in wording the goal to reach, and also elements assumed to be meaningful in the context. This can be seen as the

- The operational point of view makes up for the determination of the task. This consists in describing actions that must be undertaken to reach the goal, i.e. a recipe. The operational description of a task is held in a plan that decomposes the task into other tasks and steps. Steps are elementary actions, called inferences in KADS. A step is described as a needed input, a needed mechanism, an obtained result.

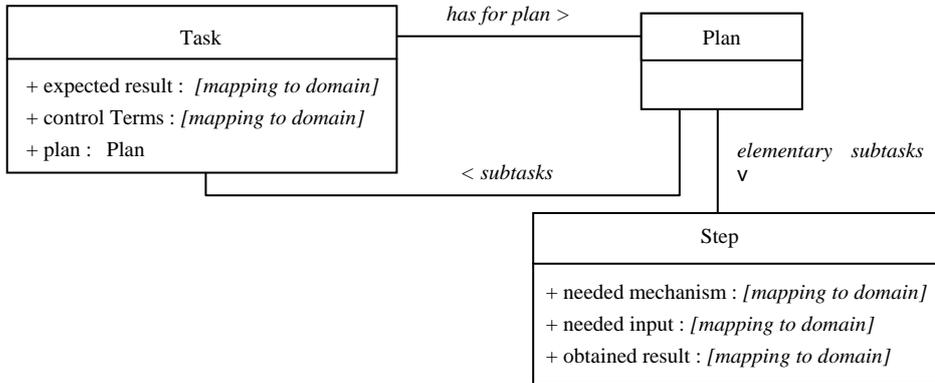


Fig. 2. : Components of a task.

### 2.2.2 The concepts held in the geographic domain

The domain holds the description of manipulated objects. These objects are what fulfils the different roles. In our work, they belong to three categories :

- Some domain elements are concepts needed to denote goals and control terms, e.g. navigation, distance city, mountain. They are often not explicit in GDBs but are still needed by users to express their intended use.
- Some domain elements represent data sets. These can be raw data. These can also be derived data, since in a process plan the input of a step might be the output from another step, i.e. derived data.
- Some domain elements represents stored mechanisms, i.e. GIS process or specific algorithms.

Component	Domain element
Expected result Control terms	Applicative Concept (ex : a location, a map)
Needed input Obtained result	Raw or derived data set (ex : vector road objects)
Needed mechanism	GIS process, specific algorithms (ex : network calculus algorithms)

Fig. 3. : Task components and elements of the domain which value them

The generic representation of a domain element is detailed very briefly hereafter. This object has four attributes :

- its name,
- a set of properties used to specify the element (attribute or link with other elements)
- a specific property : “representedBy” which links an element to other elements of the domain that are specific representations of it, typically when a concept is used to word an expected result, the obtained result will be an element representing the expected result.
- a specific property : “producedBy” which links an element to the steps and tasks that have it as goal or obtained result.

### 2.2.3 The Role concept

The concept of role is at the core of the mapping from elements of the domain to the vocabulary of the tasks. To be more precise, it is used to value an input or output of a step, and to value a goal and control terms of a task, according to the principles set fig 2.

To represent this mapping we use a specific object : the "Set".

A set has two characteristics : its intension and its extension.

The intension models the set membership conditions. In our model one type of intension has been represented so far : it is the definition of a generic element so that every element that specialises this element belongs to the set, e.g. "entity which characteristic scale is 1:10 000".

The extension is the list of the elements belonging to the sets, e.g. "France, Germany, Spain".

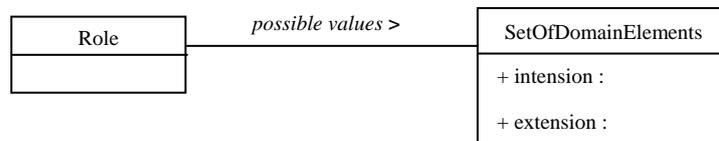


Fig. 4 : Role and Set components

## 3 ACCESS FUNCTIONALITY : TO STORE AND REUSE GEOGRAPHIC APPLICATIONS PATTERNS

### 3.1 User access

#### 3.1.1 Rough scenario

The user's need is represented in the system as a specific task, taskU. TaskU actually represents the intended use of the stored information. The user then browses both the declarative and operational aspects of taskU. These aspects are kept coherent by the system.

The browsing of the declarative side, i.e. goal and control terms, is an interactive specification process. It is the query part of the browsing.

The browsing of the operational side is the answer part of the browsing. The system determines the plan of the taskU according to the specified declarative characteristics.

The answer can be read by the user as : which static and dynamic resources he should assemble, and in what way, to build an application so that the obtained result would more or less match the expected result he has specified.

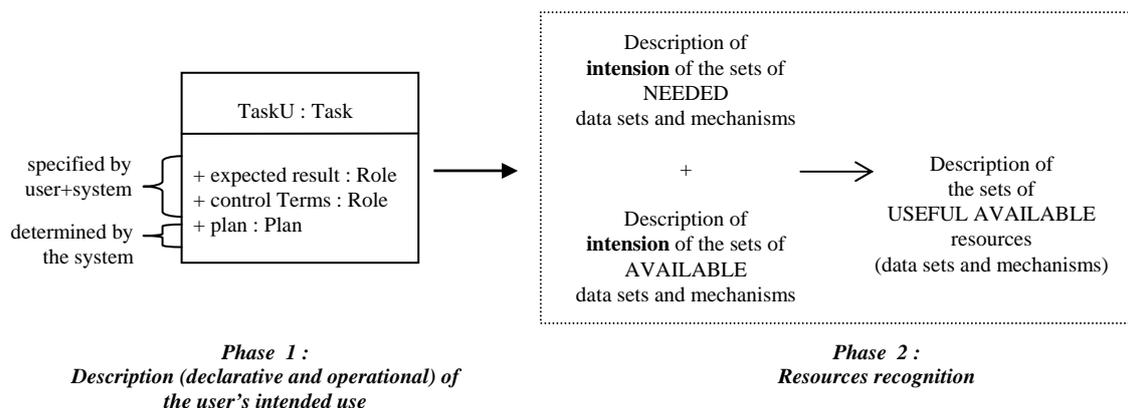


Fig. 5. : User access to resources usage : a consistent specification and determination of a task representing the user's need, i.e. its intended usage of the resources.

### 3.1.2 To store use patterns

The model is used to store known geographic application, as specific tasks, so that they can be reused following the above scenario. Task is actually an abstract java class.

The first step of the storing process consists in building a new class, e.g. Task1 extending the class Task. The components of this new class are determined as follows :

- The goal of Task1 is worded in a generic way, and mapped to elements in the domain (eventually these elements must be created and added in the domain). So far several goals have been worded : the determination of the location of an entity, the determination of the entities located at a spatial reference, the determination of possible accesses to a location, the determination of a route.
- A generic plan to perform the application is depicted in terms of generic mechanism and input and output. It is the initial value of the plan of Task1.
- What can lead to a specification of the plan is expressed as control terms in Task1, with initial values.

Two abstract methods of the class Task must be defined in Task1:

- `specifyMyself()` : when a role of the task is specified this method suggests specifications of other roles, that can be accepted or not by the user.
- `determineMyself()` : when a role of the task is specified this method specifies the plan.

### 3.1.3 To reuse stored use patterns to perform the scenario

The reuse process is actually the building of TaskU.

It first consists in the user's choosing a stored geographic task. TaskU is then defined as an instance of the class depicting this geographic task. TaskU's goal and control terms are then determined and initially valued and TaskU has two methods, `specifyMyself()` and `determineMyself()`.

The specification goes on by specifying the goal and control terms. To specify a role is equivalent to reduce the set of candidates. This can be done by specifying the intension, which means constraining the value of properties of the domain element used to describe the intension. This can also be done by reducing the extension. Each time the user specifies a role, the method `specifyMyself()` possibly proposes new specification of other roles.

Parallel to that, the determination process is led by `determineMyself()`. Each time the user specifies a role, this method tries to specify the plan , i.e. to specify or decompose the mechanism with its inputs and output.

Fig. 6 shows the java frames supporting the specification process and the display of the determined plan. On this figure, one frame leads the specification the task. In this case, two main roles are identified *localisation* (location), and *entité à localiser* (entity to locate). The tree panel shows items to characterise the intension of the value of the role. The panel besides the tree is used to suggest new specification by the method `specifyMyself()`. The small frame shows the operational description of the task, it is updated by the method `determineMyself()`. The bottom panel depicts possible actions for the user : to specify, to save or not this state of specification, to see if the resources are available as well as their characteristics.

### 3.1.4 A browsing process

During the session, the user can save the state of specification of TaskU. It is saved in a history of the session. The user can browse this history, which is displayed in a tree. He can start again the specification process from any step of the history. This allows him to compare several tasks.

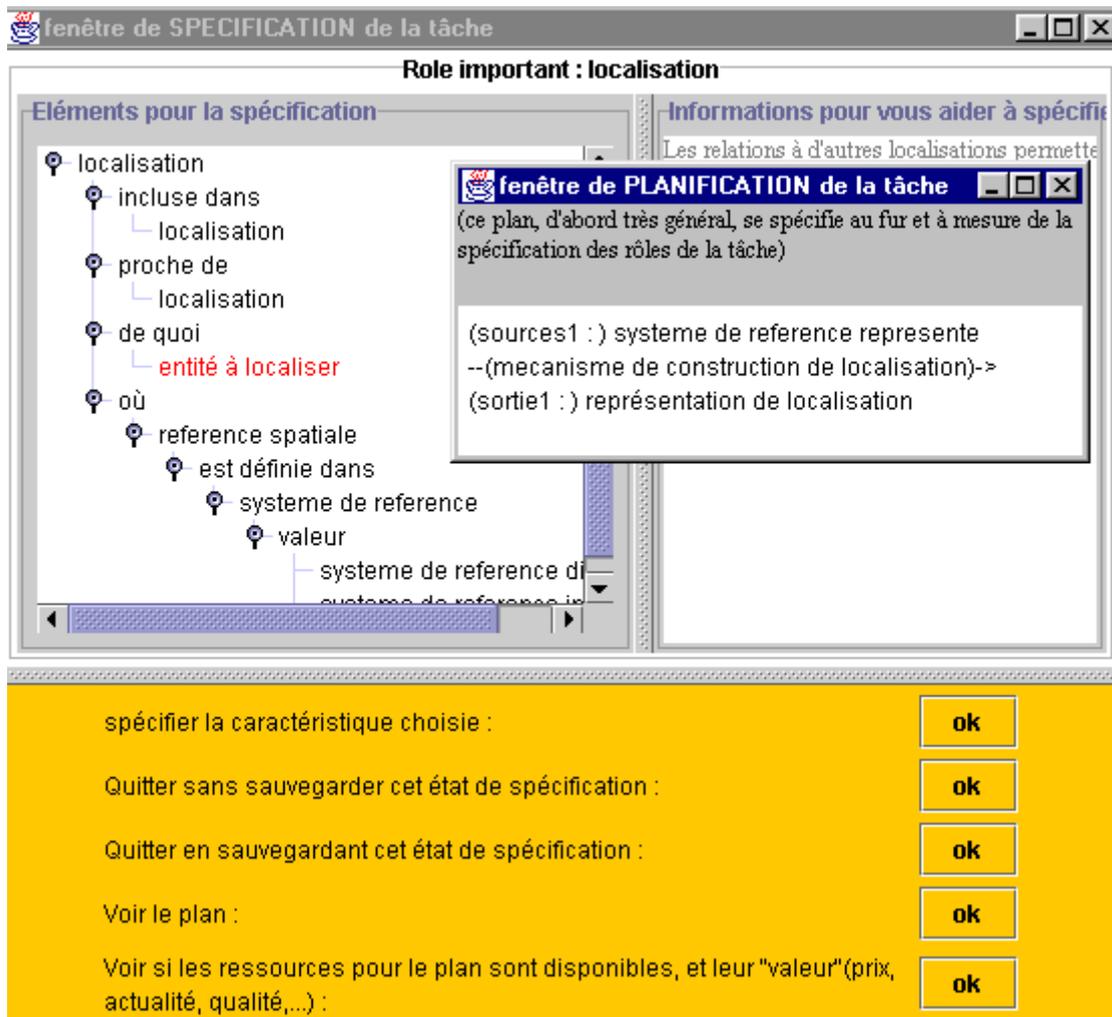


Fig. 6. : Some frames of the prototype.

#### 4 CONCLUSION AND PERSPECTIVES

Users' access to geographic information resources is supported here as a focused exploration of metadata describing the set of applications that can be built upon available geographic data : neither the name of resources nor its content is searched, but instead the possible application objectives that can be achieved using such resources. Again, the system does not derive knowledge on demand but rather it stores and proposes customisable geographical knowledge derivation recipes.

This is made possible thanks to a specific task-based model : an object-oriented task-based model derived from a well-known task-based model in AI, CommonKADS. This model describes the GDBs, the tools to handle the data, the methods to handle the tools, and the concepts needed to word application objectives.

A prototype tests the following functionality : the incremental specification by the user of a particular application and its parallel planning by the system. We also intend to build other functionality :

- The enriching of the knowledge base is an important functionality. Indeed we do not aim at building an exhaustive knowledge base but rather a theoretically sound framework, the generic model and the first instances, that should further integrate other instances describing existing data, processes, methods and application objectives.
- It is important for users to explore in a bottom-up way the knowledge base of the system, i.e., given a

In the prototype, the interface is not intuitive to the user. This issue is dealt with in another work at the COGIT laboratory. Hubert integrates advance means of communication, e.g. natural language and the use of samples, to support a real dialogue between a user and a geographic information server system [Hubert 01].

## 5 ACKNOWLEDGEMENTS

The author wishes to thank Didier Richard, Sébastien Mustière, Anne Ruas, Cécile Lemarié and Jean-Philippe Lagrange from IGN for fruitful discussions, as well as Nathalie Aussenac-Gilles from the IRIT Institute for her helpful comments on this approach and Marinette Revenu and Valérie Ficet from the University of Caen for their advises during the modelling process.

## 6 REFERENCES

- [Balaguer et al. 97] F. Balaguer, S. Gordillo, F. Das Neves, Patterns for GIS Applications Design, in proceedings of Pattern Languages of Programming, PLoP'97, Monticello, Illinois, USA, 1997
- [Bishr 97] Yaser Bishr, Semantic Aspects of Interoperable GIS, PhD Thesis, ITC, Netherlands, 1997
- [Bucher 00] Bénédicte Bucher, Users' access to geographic information resources : a model of tasks and roles to specify intentional uses regarding available resources, in *Geographical Domain and Geographical Information Systems*, Winter, S. (Ed.), GeoInfo Series Vol. 19. Institute for Geoinformation, Vienna University of Technology, Vienna, Austria, 2000, 122 pp.
- [Chandrasekaran et al. 98] B. Chandrasekaran, J. R. Josephson, V. Richard Benjamins, The Ontology of Tasks and Methods, in Proceedings of KAW'98, Eleventh Workshop on Knowledge Acquisition, Modeling and Management, Alberta, Canada, April 1998
- [Corona and Winter 01] Barabara Corona, Stephan Winter, Navigation information for pedestrians from city maps, in *GI in Europe : Integrative, Interoperable, Interactive, proceedings of the 4<sup>th</sup> AGILE conference*, april 2001, Brno, Czech Republic
- [Ficet et al. 99] V. Ficet-Cauchard, M. Revenu, C. Porquet, *Knowledge Management in Image Processing by Means of Case-Based Reasoning*, Workshop on Knowledge Management and Organizational Memories, IJCAI, Stockholm, Sweeden, 1999
- [Gomez and Benjamins 99] Asuncion Gomez , Richard Benjamins, Overview of Knowledge Sharing and Reuse Components : Ontologies and Problem-Solving Methods, IJCAI, Stockholm, 1999
- [Guarino et al. 99] N. Guarino, C. Masolo, G. Vetere, Ontoseek : Content-Based Access to the Web, IEEE 1999
- [Hubert 01] Frédéric Hubert , Assistance Mechanisms for Use Needs Specification in Geographical Information on the Web, in *Proceedings of the 20<sup>th</sup> International Cartographic Conference*, Beijing, China, 2001, to be published
- [Kuhn and Frank 97] Werner Kuhn, Andrew Frank, The Use of Functionnal Programming in the Specification and Testing Process, interop97
- [Rhind 01] David Rhind, Global and National Geographic Information Policies, Practice and Education in a g-business world, in *GI in Europe : Integrative, Interoperable, Interactive, proceedings of the 4<sup>th</sup> AGILE conference*, april 2001, Brno, Czech Republic
- [Schreiber et al. 99] A. Schreiber, J. Akkermans A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde, B. Wielinga, Knowledge Engineering and Management, The CommonKADS Methodology, MIT Press, 1999
- [Schreiber et al. 00] G. Schreiber, M. Crubezy and M. Musen, A Case Study in Using Protege-2000 as a Tool for CommonKADS, in proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management, EKAW'2000, Juan Les Pins, France, 2000
- [Vckovski 98] Andrej Vckovski, Interoperable and distributed processing in GIS, PhD Thesis, University of Zurich, ed. Taylor & Francis, 1998