Proceedings of the 6th AGILE
April 24th-26th, 2003 – Lyon, France

TRANSLATING USER NEEDS FOR GEOGRAPHIC INFORMATION INTO METADATA QUERIES

Bénédicte Bucher

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

1. INTRODUCTION

1.1 Geographical data sets retrieval and metadata querying

Geographic metadata models are designed in particular for cataloguing purposes [4] [6]. To that effect, a formalism for querying metadata architectures, which scope includes resource discovery, is proposed in [7]. In spite of this, users formulating metadata queries to retrieve data sets answering their need is still a difficult process. The need expression is seldom naturally decomposed into metadata elements such as resolution, spatial extension and schema. To analyse this issue of users formulating a metadata query, we decompose their need expression after the three following aspects : implicitly, *a user needs data that provide a representation of geographic features that are input for the derivation of a specific result* like an itinerary, a map, or the geo-referencing of his own data. The explicit formulation of his need by the user will stress more or less one of these three aspects :

- needed data,
- needed representation of geographic features,
- needed application result.

We list, in this introduction, obstacles to the formulation of an efficient metadata query; which are linked to the expression of an aspect or the other.

If the need is expressed solely as a need for data, i.e. as constraints on properties of the searched data, several obstacles may occur. The properties relevant to the user may be different from the metadata elements. For instance, the user may be interested by the actual exhaustiveness of the data. This property is not a metadata element but its value can be inferred from the values of the elements date -or temporal extent- and completeness. Moreover, even if the properties used in the user query are those used in the metadata description, the user may have a specific vocabulary to express constraints on these properties different from the vocabulary used to document the metadata. For instance he may look for aerial pictures which spatial extent "includes his house".

If the need is expressed as a need for represented geographic features, the user should specify what objects of reality he wants to have represented and what representation of these features he wants. Such a query cannot always rely on the metadata element "schema" because of the lack of agreement on semantic categories used to represent geographic space, as detailed by [9].

Very often, the need is expressed as a need for an application result because user concepts, such as "flood bed", are not explicit in the raw data but are to be derived by application of existing GI functions on the data. The user is seldom able to decompose his need into a need for data and a need for functions.

1.2 Providing a formal solution to the metadata querying issue

Dealing with these obstacles is not always difficult for human experts, if they do it on specific cases –on specific resources for instance-. They only have to know some relevant items. Knowing metadata elements and their meaning, as well as the user terms to express constraints on these elements, allows experts to translate user data need into a metadata query. Knowing user concepts to represent geographic space and GI data concepts allows experts to translate user needed representation of geographic features into a metadata query. Knowing geographic functions, data to which they apply, and possible application results that can be calculated with them allows experts to translate user need for application result into a metadata query.

But a more formal solution is required for end users to make the most of the huge ongoing resource discovery metadata effort. Indeed, the translation would then not be dependant from a specific expert and limited to specific resources. Such a formal solution requires formal elements corresponding to the above listed varied expertness:

- a model to map data properties that are relevant to users with data properties that are documented in metadata,
- ontologies of geographic information,
- a formal model that supports the following functions : indexing existing geographical data, indexing existing geographical functions, expressing a user need, matching a user need expression with metadata queries.

These are complementary elements.

In this paper, we focus on the last element. Bricks for a model supporting these functions already exist. Indexing existing geographical data is dealt with in spatial data infrastructures. Indexing existing geographical functions is an emerging process. Supporting the expression of the user need is a complex knowledge engineering issue. The need expression is actually need dependant. For instance if someone wants to calculate an itinerary, he may specify his departure place and his type of vehicle. The mapping of this expression with a metadata query is also need dependant ; there is no known function to match expected results with needed data. More precisely, doing this mapping is not that difficult for some human experts but it is very difficult to do it in a formal way.

2. OUR APPROACH

In this section we introduce the use of tasks and roles to formalise the matching between a user need -specified as an expected application result- and the data and functions that calculate it –specified as metadata queries-.

2.1 Tasks and roles-based models

Models of geographic information exist but none of them is dedicated to all the above required functions. To propose such a global model, we do not rely on geographic information theory. We use lessons learned in the technical field of knowledge representation to support reasoning, and more specifically to support knowledge sharing and reuse. Indeed, we wish to enhance a specific knowledge reuse : the design of applications meeting user needs with available geographical data.

In this field, a reasoning is classically represented as a problem-solving process, which entails two different components : the description of the problem and the description of how to solve it.

The concept of *task* is introduced to represent a family of problems. It is composed of several components [3]:

- a description of the context (initial and final states) that can be specified to represent the specific descriptions of the problems belonging to the family,

- a resolution to reach the final state that can be specified to represent the specific resolutions of the problems belonging to the family.

Importantly enough, the process of specifying a task into the description if a problem belonging to its family is usually a step by step process of the form : a specification of the context implies a specification of the resolution. For instance if a task consists in drawing a cartographic presentation, its description entails the element "input geographic data", and its resolution will vary if this elements is specified in the problem context to be raster data or to be vector data.

Another concept introduced in this technical field of knowledge representation is that of *role* [5]. It is a variable which name refers to the part played by this variable in the process where it is defined, e.g. a classification criteria.

Roles are used to describe tasks contexts. Specifying a task description into a problem description implies to specify the roles in the task description, which means reducing their sets of possible values (**Erreur ! Source du renvoi introuvable.**). In the following, we sometimes use the expression "task totally specified" instead of "particular problem". Since task specification is a step by step process, a task can be partly specified into a less generic task. For instance, “route determination” can be specified into “route determination for a vehicle equals to a bus”.

Roles also appear in the description of task resolution. If the task is decomposed into subtasks, these are the roles composing the context of these subtasks that are not explicitly represented in the context of the upper level task. For instance, the role "input geographical data" does not appear explicitly in the context of the task "route determination", but it appears in its resolution.

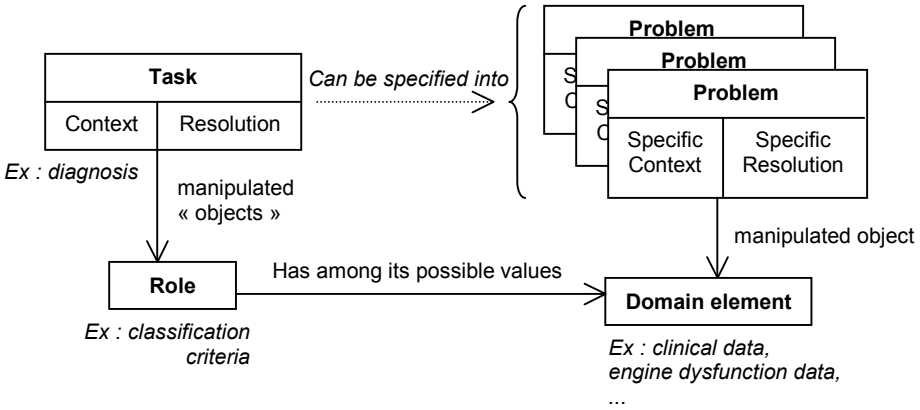

Fig. 1 The use of concepts of task and role to describe a family of problems.

Let us make a parallel with the domain of theatre to illustrate these modelling constructs and their relevance in our context. Resources are available actors. In this parallel, a user need is a result that can be so to say derived from the resources : a specific realisation of a specific piece of theatre. It can be expressed for instance as "a classical tragedy with a female main character".

A task can be defined for instance as follows: the possible realisations of a given piece of theatre, e.g. "the realisations of Phèdre".

The generic description of the task comprehends the piece livret. This relies on roles corresponding to the piece character, e.g. the role "Hippolyte" which set of values is defined by the profile of actors that can potentially play this character.

The generic resolution contains, among others, mechanisms to chose a specific actor for each character. These choices depend not only on the actors profiles but also on elements like the available funds to pay actors. These elements of the context which specification acts upon the resolution should also be represented as roles in the generic description, like "available funds", "size of the theatre".

2.2 Application to matching user needs and metadata

Modelling geographic tasks supports the functions listed in the preceding section as follows (**Erreur ! Source du renvoi introuvable.**).

High level tasks describe families of geographic applications such as "to draw a map", "to locate something", "to determine an itinerary". The functions are depicted as elementary tasks, e.g. to build a buffer, to overlay representations. All these tasks have some roles dedicated to describing input data. The possible values for these roles are geographical data sets, described by their metadata.

The expression of the user need is supported by the task to which context his need is close to. The specification of the user need is actually the specification of the context of this task into the description of his need. The specification of the task context triggers the specification of its resolution. The resolution of the user problem entails some roles describing needed input data.

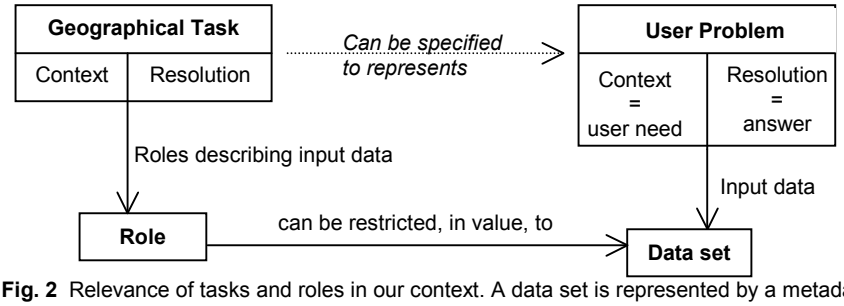

Fig. 2 Relevance of tasks and roles in our context. A data set is represented by a metadata selection.

We underlined that the specification and answering of a need were need dependant (sect. 1.2). This is rendered in a tasks model : knowledge about how to specify a task is part of the task description.

This kind of model cannot be exploited directly by end users. It must be encapsulated in a cooperative problem-solving environment, or, in our context, a cooperative geographic application design environment.

3. TAGE : A MODEL OF GEOGRAPHIC TASKS AND ROLES

In this section we present our tasks based model : TAGE ("tâches géographiques").

3.1 Main elements

To build a tasks and roles based model we rely on the expertise model proposed in the ESPRIT project KADS [10]. This is a reference model in knowledge representation. Since it is quite complex, and not operational so far, we did not use it as such. An initial version of our model has been introduced in [1]. We summarise hereafter the main elements of this model.

A *role* is composed of a name and a variable *candidates* which datatype is *set*. A *set* is an object encapsulating different definitions of a set, intensive or extensive, and associated methods like restricting the set by specifying its intensive definition or by selecting an item inside its extension (these restricting methods are named *specify*).

A task is composed of a context, a resolution, and specification knowledge to infer descriptions of contexts and resolutions of problems belonging to the task family.

The context is composed of *roles* : the *inputs* and the *output*. For instance the task "to locate something" has for *output* a role named "spatial reference" and for *inputs* roles named "entity to locate" and "known location information". The output "spatial reference" has for candidates the set of everything that can be a spatial reference in a user reasoning : coordinates in a formal spatial reference system (direct or linear), item in an administrative partition, symbol on a map, route indications.

To describe resolutions of tasks is often difficult, and to define a representation item to formalise all task resolutions is all the more difficult. Indeed to a generic context does not always correspond an obvious generic resolution. For instance there is no obvious generic method for the task "to locate something". Like [10], we represent tasks which resolution varies much among the problems of the family by the class *complex task*. The generic resolution of a *complex task* is represented by an object *method*, which also encapsulates the specification knowledge of this resolution. We represent tasks which resolution does not change among the problems of the family by the class *primitive task*. Its resolution is represented by an object *mechanism*.

3.2 The element *method*

To define a *method*, we consider that the resolution of a problem is a derivation plan, represented as a structure of primitive tasks totally specified, i.e. which roles have only one specific value. Thus a *method* should support the production of all possible derivation plans corresponding to a given family of problems (

). We structure it as a generic plan plus specification rules.

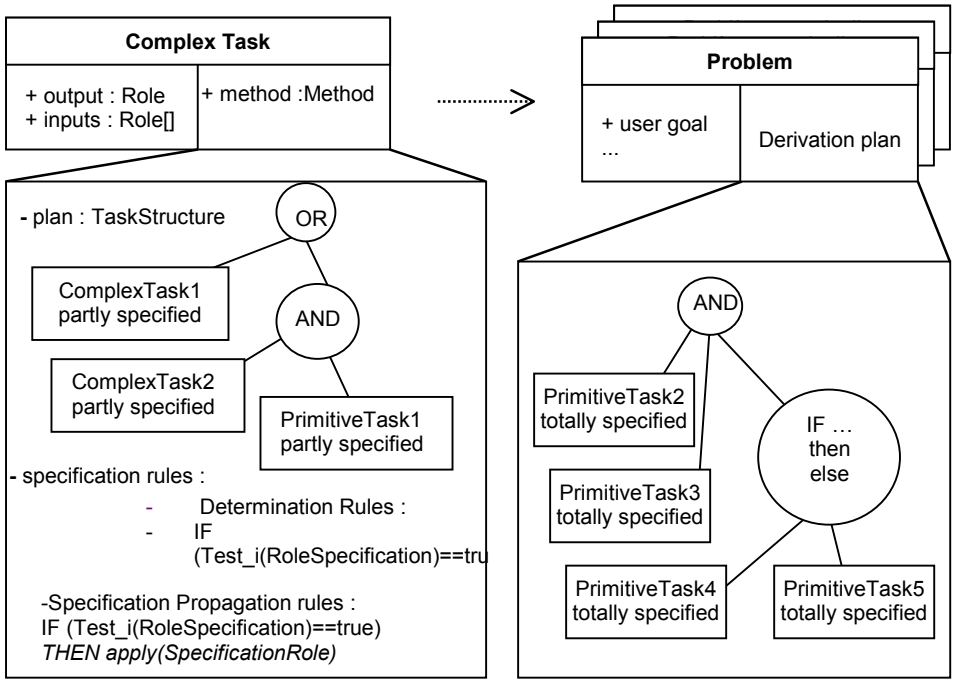

Fig. 3 Representation of the resolution of a complex task in TAGE.

The generic plan is also a structure of tasks but different from a derivation plan. Its elements are tasks, primitive are not, partly specified and its conjunctions may be "OR". This will be illustrated below.

The specification rules entail the knowledge to infer derivation plans from the generic plan, step by step, according to the current specification of a role (input or output) of the task. A *TaskStructure* has a specific attribute called *vocabulary* which associates terms with roles in the structure.

The generic plan we propose for the task "to locate something" is a "OR" structure with sub-structures ss1 and ss2. Ss1 is a "AND" structure of four partly specified complex tasks (**Erreur ! Source du renvoi introuvable.**) :

- CT1 is the task "to acquire a spatial reference system", which output is specified to be a direct or a linear spatial reference system (a GIS-readable reference system).
- CT2 is the task "to produce a spatial reference", which input "spatial reference system" is specified to be the output of CT1.
- CT3 is the task "to acquire a spatial reference system".
- CT4 is the task "to produce a spatial reference", which input "spatial reference system" is specified to be the output of CT3 and input "known reference" is specified to be the output of CT2.

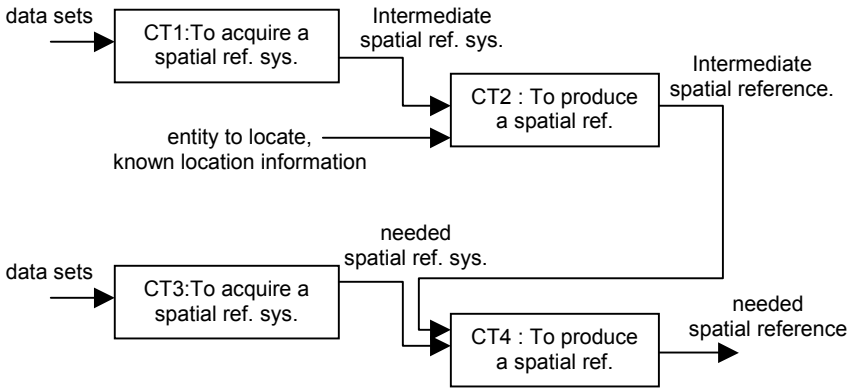

Fig. 4 First sub-structure of the generic plan of the task “to locate something” in TAGE. Boxes are partly specified tasks and the arrows are roles.

The overall strategy of ss1 is twofold. CT1 and CT2 produce a first spatial reference of the entity by exploiting every clue the user can provide about this entity and its location. CT3 and CT4 translate this reference into a spatial reference in the spatial reference system the user wants, which can be not only direct or linear reference systems but also a map or a route.

Ss2 is a "AND" structure of two partly specified tasks duplicating the beginning of ss1, let us call them CT5 and CT6. The sub-structure ss2 corresponds to derivation plans when there is no need for an intermediate spatial reference system, for instance because the user already has a GIS-readable spatial reference system of its entity, or because he wants such a reference.

The vocabulary of the overall structure is initially composed of four terms : intermediate spatial reference system, intermediate spatial reference, needed spatial reference system, needed spatial reference. The term "intermediate spatial reference system" is associated with the output of CT1 and the output of CT5. The term "needed spatial reference system" is associated with the output of CT3 and the output of CT5.

Specification rules are of the form : IF (the entity is specified to be of nature road) THEN (role "data input" of task which output is "intermediate spatial reference system" should entail in its schema the element road). Another rule is for instance : IF (the known location information is a spatial reference in a GIS-readable system) THEN ((chose substructure ss2) AND (the input "known reference" of CT5 is specified to be the known location information)).

To evaluate the expressiveness of this model, we instantiated it to represent different types of tasks : *intentional tasks*, like "to locate something", which representation focuses on the expression of the user need, i.e. on the representation of relevant elements in the context, and *functional tasks*, like "to match one data set with another", which representation focuses on the accurate description of the method. It soon appeared that being a complex or a primitive task is seldom an intrinsic property of a process. It is rather a question of granularity of the task model. We have chosen to consider geographical functions, like GIS operations, as primitive tasks of our model.

4. THE IMPLEMENTATION

4.1 Prototype 1: a tasks server

A first prototype of a cooperative problem-solving environment has been implemented in java [2]. Its objectives are to evaluate :

- the encoding of patterns of geographic applications as tasks, roles and methods. The tasks "to locate something", "to aquire a spatial reference system" and "to produce a spatial reference" are implemented in this prototype.
- the encoding of the dynamism of the cooperative problem-solving environment. This dynamism relies on an interface to interact with the user and a core engine that exploit the knowledge base to specify and answer the user need.

The strong limitations of this prototype are the following :

- the user interface is not very friendly,
- we lacked metadata to index data sets or GIS functions in this prototype so that proposed derivation plans remain relatively abstract.

4.2 Prototype 2 : a metadata server

To address the second limitation, a prototype of metadata server was built in the MSc research project of Ali Taouss. Acquiring metadata was the toughest part of this work. They were restructured in a model close to ISO19115 and stored in a Oracle spatial database. The user can query the server through a Web interface. He specifies metadata elements like "spatial extent" or "keywords". We made use of some contextual databases similarly to the classical exploitation of gazetteers in geolibraries to specify user spatial area of interest. These contextual databases are :

- BDAdmin®, an administrative database. The user can search for data sets spatially related to a given administrative entity, e.g. "vector road network in la région Provence".
- Route120®, a small scale topographic database. The user can search for data sets spatially related to large geographic objects, e.g. "1:25 000^{e} maps along the river Seine".

To make use of these contextual DBs, the notion of *complex value* was introduced in the query model, similarly to *synsets* in the thesaurus wordnet. A synset is a group of synonymous terms. A complex value is a group of synonymous designations of an item, where a designation can be a value or a mechanism to produce this value (Fig. **Erreur ! Source du renvoi introuvable.**).

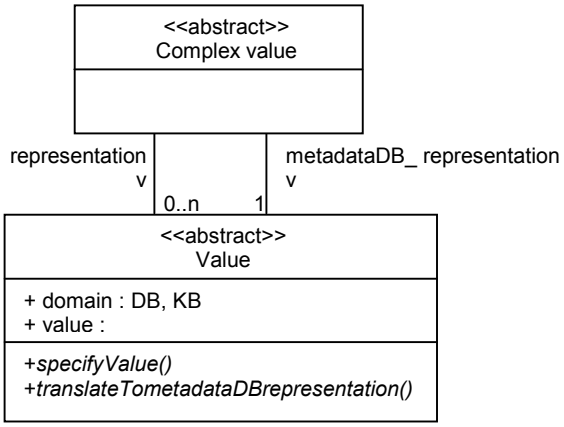

Fig. 5 Classes used to group the different representations of a value. This abstract class is specialised to represent "complex values" such as Keyword or Spatial_Area.

4.3 Prototype 3 : TAGIN

The ongoing prototype, called TAGIN, is developed to bring together the two first prototypes, and to insist on the interoperability of this solution.

Bringing together the two first prototypes relies on defining the *candidates* for an input data role by a specific intention which is a metadata query (**Erreur ! Source du renvoi introuvable.**). Still, we have to work on a model for mapping data sets properties that are relevant to a user -more precisely to tasks- with data sets properties appearing in metadata.

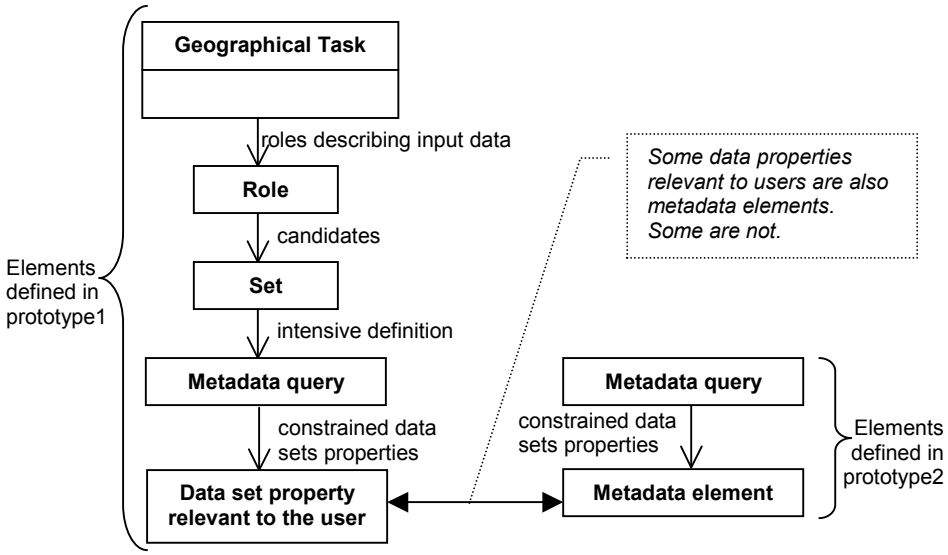

Fig. 6 Bringing together prototype1 and prototype2 in the TAGIN prototype.

Compatibility with existing standards is a crucial requirement for this prototype.

Concerning data descriptions and metadata querying, we follow the lines of the ISO19115 model for geographical metadata and of the OpenGIS specification for querying metadata architectures [4][7].

Concerning services description, ISO19115 is rather poor about this type of resources. It entails relevant elements, like quality, that are to be specified to describe services. We are also awaiting the definite ISO19119 model for geographic services which should provide services metadata.

Concerning the domain description, GML supports the description of the GIS readable domain [8]. Some elements of the domain, that are to represent user concepts, still have a representation specific to our system and whereas they should be represented in an agreed on ontology.

The TAGE model was carefully reviewed to ensure the shareability of tasks.

We put many efforts on refining the class *Method* so that it does not contain procedural knowledge. In particular, all specification rules, which include sets restriction operations, are represented as declarative variables. This relies on specific classes used to write rules like SetSpecification that describes all possible ways of specifying a set thanks to simple type variables. Tasks can now be encoded in XML documents, thanks to a set of XML Schemas.

Particular tasks are modelled as reusable elementary components. This principle is illustrated in the task “to locate something”. It reuses two tasks, “to acquire a spatial reference system” and “to produce a spatial reference”. Each of these tasks is defined independently from the location task and can be reused in the plan of whichever task. Reusing a task t1 as a subtask in the plan of a task t0 is performed through specifying t1 context (**Erreur ! Source du renvoi introuvable.**). Never can the resolution of t1 be directly specified in t0 method. The specification of t1 resolution is always triggered by the specification of t1 context in t0 method. Every specification knowledge of a task resolution is encapsulated inside its own method so that its behaviour is determined only by the specification of some of its roles.

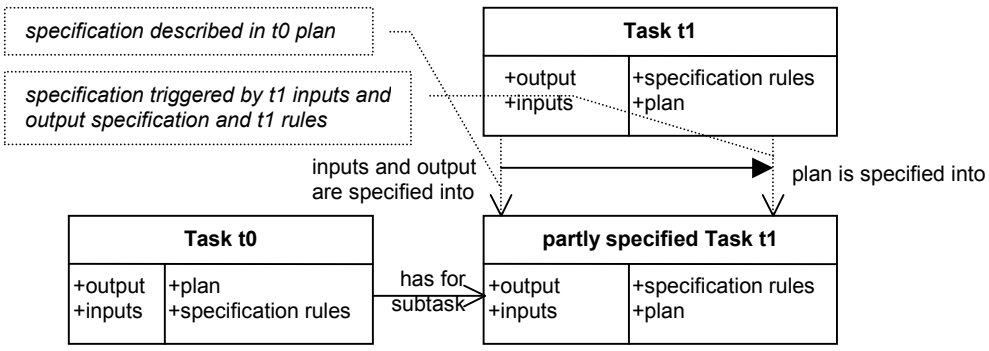

Fig. 7 Describing tasks as reusable components : the specification of a task resolution is always triggered by a specification of its context, even if it is reused in the plan of another task.

5. CONCLUSION

Expressing their need as queries on properties of raw geographic information resources, i.e. as metadata queries, is often too complex for users. Solve this issue for one user is not that difficult, but solving it for every possible users is a complex challenge. Indeed it implies to formalise some know-how involved in this mapping : how to use geographic information raw resources to derive meaningful information for users.

We propose a model to represent this knowledge and to encode it in XML documents, TAGE. TAGE is based on tasks, roles and methods. A cooperative problem-solving environment is associated to TAGE : TAGIN. TAGIN supports the user-system cooperative

specification of tasks to represent the user problem, i.e. his need and the answer to this need.

This work was initiated to support user access to data sets. Its scope is now enlarged to supporting user access to several types of geographic information, not only data sets but also functions and methods.

6. ACKNOWLEDGMENTS

The author wishes to thank Didier Richard, in charge of the data diffusion project at IGN, for fruitful discussions and various collaborations, as well as the CONSUL team of the COGIT among which this work was lead.

7. REFERENCES

[1] Bénédicte Bucher, A Model to Store and Reuse Geographic Application Patterns, in *GI in Europe : Integrative, Interoperable, Interactive, proceedings of the 4th AGILE conference*, april 2001, Brno, Czech Republic

[2] Bénédicte Bucher, Structuring and enriching metadata to enable users' access to geographic information resources, in *proceedings of the 20th International Cartographic Conference*, Beijing, China, august 2001, vol 4, pp. 2791-2797, 2001

[3] Chandrasekaran B., Josephson J., Benjamins R., "The Ontology of Tasks and Methods", in *proceedings of KAW '98*, 1998

[4] ISO/TC211, *Text for 19115 Geographic information - Metadata for registration,* ISO/TC 211 draft document for DFIS, october 2002

[5] Marcus S. (Ed.), *Automating Knowledge Acquisition for Expert Systems*, Boston: Kluwer Academic, 1988

[6] Open GIS Consortium Technical Committee, *The OpenGIS ® Guide : Introduction to Interoperable Geoprocessing and the OpenGIS Specification*, Kurt Buehler, Lance McKee (Eds), 3rd edition, June 3, 1998

[7] Open GIS Consortium Technical Committee, *The OpenGIS® Catalog Services Specification,* Douglas Nebert (Ed), december 2002

[8] Open GIS Consortium Technical Committee, *The OpenGIS® Geography Markup Language (GML) Implementation Specification,* S. Cox, P. Daisey, R. Lake, C. Portele, A. Whiteside (Eds), january 2003

[9] Jonathan Raper, Unsolved problems of spatial representation, in *proceedings of SDH'96*, 1996, pp.14.11-4.11

[10] Schreiber A., Akkermans J., Anjewierden A., de Hoog R., Shadbolt N., Van de Velde W., Wielinga B., *Knowledge Engineering and Management, The CommonkADS Methodology*, MIT Press, 2000