A Spatial Data Model for Navigation Knowledge

Claramunt Christophe (1) - Mainguenaud Michel (2)

(1) Swiss Federal Institute of Technology
Rural Engineering Department
Spatial Information Systems
CH1015 Lausanne, Switzerland
Tel: + 41 21 693 57 83 - Fax: + 41 21 693 57 90
Email: Christophe.Claramunt@dgr.epfl.ch
Url: http://dgrwww.epfl.ch/SIRS/index.html

(2) France Telecom
Institut National des Télécommunications
9 rue Charles Fourrier
F91011 Evry, France
Tel: + 33 1 60 76 47 82 - Fax: + 33 1 60 76 47 80
Email: Michel.Mainguenaud@int-evry.fr
Url: http://www-inf.int-evry.fr/BasesDeDonnees/Cigales/cigales.html

***Abstract**: This paper presents a spatial data model suitable for the representation of navigation knowledge. The model analyses a navigation process from a cognitive point of view, with the visual environment necessary for its comprehension. The decomposition is based on the spatial view concept, defined as a logical, dynamic representation of spatial data. The model allows the coexistence of several abstraction levels. A language and operators define and specify the model.*

***Key words**: navigation knowledge, cognitive collage, spatial collage, spatial view.*

I. Introduction

Most current data models describe space in a cartographic, continuous form [e.g. Bur 86, Peu 88]. However, other perceptions of space, narrative or navigational, are also used. This is the case of navigation knowledge, which corresponds to the mental description of a spatial displacement. This perception is called a cognitive map [Kui 78]. Its representation is closer to human thinking than a static map. Cognitive maps, largely studied in psychology and linguistics [Mar 91], inherit the diversity and richness which characterise human thinking but also possess its imperfections. The space represented by a cognitive map has four dimensions (three for space and one for time). Its spatial representation is dynamic (cohabitation of different spatial perceptions) and discontinuous (partial knowledge), and thus less easily identifiable by current spatial data models. The perceived topology is relatively well preserved and essentially linear; notions of distance are, on the other hand, approximated. These properties show the variability of the memory expressiveness when displacement is perceived. In fact, a navigation process comes from a mental memory of our sensory activities. This allows the constitution of a cognitive displacement memory whose quality will vary according to its users and will decrease over time.

The objective of this study is to model a cognitive map representing navigation knowledge (a model of a model). Formalisation of this cognitive representation allows the preservation of these process memories and extends their application potential. Identification of decision criteria during the realization of a navigation process [Gol 95] or of network algorithms will not be developed [Ken 80]. The proposal defines a model and operations that allow decomposition of navigation knowledge and its presentation as a form of dynamic cartography. The model identifies element characteristics which enable the description and presentation of these processes. It uses the concept of spatial view defined as a dynamic and flexible tool for spatial data representation [Cla 94].

Section II describes the principles of navigation knowledge; section III explains the contribution of the spatial view; section IV proposes a framework for representing navigation knowledge; manipulation principles of the model are presented in section V; an application illustrates the model in section VI and a conclusion is formulated in section VII.

II. Navigation knowledge principles

Navigation knowledge is provided by the application of a mental model which depends on a user perception [Tim 92, Tho 93]. It is the result of an experience if it is known, or it has to be experimented if it acts as a proposed solution. Navigation knowledge is defined by a **procedural knowledge** [Tho 93], the related geographical environment which provides **survey knowledge** [Tho 93] and the **perceptual space** [Che 89]. The procedural knowledge describes the spatial displacement process; the survey knowledge situates it. Perceived elements during a navigation process give visual or symbolic landmarks (e.g. photos or symbols associated with a displacement point). Each form used in a navigation knowledge environment has a specific role: survey knowledge and procedural knowledge own a spatial structure, perceived elements provide poor spatial structure but more meaning. This combination of different expression models gives global coherence to a navigation knowledge representation [Lyn 60]. The language used to describe a navigation process can be textual, gestural or graphic. Language expressions are explicit forms of navigation knowledge. They increase the expressive power of the model (e.g. from the airport, take the highway to the EPFL).

A navigation knowledge representation is based on a **hierarchical** and **incremental** cognitive model [Maa 93]. The hierarchical characteristic allows an incremental ordering of complementary abstraction levels, from strategic to specific according to Kuipers [Kui 78], from basic to secondary according to Maaβ [Maa 93] and from planning to action according to Timpf [Tim 92]. These different abstraction levels used for the description of a navigation knowledge form a **multidimensional** spatial set [Sto 93]. The semantic association of these different abstraction levels is fundamental to ensure the continuity of the cognitive process. The mental association of these multidimensional spaces is expressed by the **cognitive collage** metaphor [Tve 93]. A cognitive collage is defined as the semantic association of different data abstraction levels. A **connection** extends the map collage concept to semantic spatial associations between spatial entities which are members of the represented spaces [Ren 95]. The cognitive collage, when applied to navigation knowledge, makes necessary the identification of a form of cohabitation between different representation models, coordination that few research activities develop [Pau 89]. The goal of the cognitive collage is to associate various representation models designed with different concepts: (1) hierarchical and incremental navigation knowledge representations (2) heterogeneous geographical spaces (several abstraction levels and different spatial structures).

III. Spatial view model

The description of navigation knowledge components is based on the proposal of a spatial view, which is defined as a dynamic and flexible form of spatial data representation [Cla 94]. The spatial view is expressed as an extension of the classic database view as defined within the database context. This allows the definition of an external schema, derived from the logical level, while integrating dynamics by the application of operators [Cla 95]. External schemas provide flexibility for the representation of the spatial models which describe navigation knowledge. The dynamic character allows the identification of relevant data in each represented space. A spatial view is defined by a partially ordered atom set.

A **spatial view atom** is described by a name (Name), one or several spatial queries (Spatial_Query) operating on collections (Collection) and visualization operators (Visualization_Op). It presents a coherent spatial data set at the abstraction and scale levels while providing independence and dynamics (i.e. operators). A spatial query is defined from collections (a collection represents an entity set), and from spatial and non spatial operators which are applied to these collections. Spatial collections describe spatial entities, perceived and symbolic elements.

A **spatial view** is identified by a name (Name). The order attribute allows management of the role of each atom that composes the spatial view (Order). In the context of navigation knowledge, the order attribute distinguishes the spatial view atoms which define the navigation knowledge: the procedural description (Essential attribute, e.g. the trace of a navigation knowledge), geographical spaces (Important attribute, e.g. the network support of a navigation knowledge) and perceived elements (Useful attribute, e.g. the geographical context of a navigation knowledge). This classification defines the spatial view semantics, thus providing a criterion for managing visual conflicts related to spatial view displays and manipulations. Moreover, description of the navigation knowledge elements by a common concept, the spatial view, provides a homogeneous structure for the model. The proposed model is expressed using the notation of complex objects [Adi 87]. This notation allows the extension of classic data types. This is suited to the definition of complex data. The constructors used are aggregation, set and list (respectively noted [ ],{ } and ( )). The definition of spatial view atoms (SVA) and spatial views (SV) follows:

SVA_type = [Name: string,
Collection: {string} ,
Spatial_Query: (string) ,
Visualization_Op: {string}]

Order_type = [Essential: {SVA_type},
Important: {SVA_type},
Useful: {SVA_type}]

Elt_Manipulation_Scale_type = [Min_Limit: float,
Scale_Reference: float,
Max_Limit: float]

Manipulation_Scale_type = { Elt_Manipulation_Scale_type }

SV_type = [Name: string,
Order: Order_type,
Manipulation_Scale: Manipulation_Scale_type]

The manipulation scale of a spatial view (Manipulation_ Scale) gives the scale at which a user is situated for the manipulation of spatial view atoms [Cla 94]. The manipulation scale is defined by a numerical interval scale (float number type) corresponding to the relevant abstraction level. This interval is centered on a scale reference (Scale_Reference) and two scale boundaries (Min_Limit and Max_Limit). The users's scale position indicates which spatial primitive will be used by a spatial operation [Woo 95]. This scale acts as a filter by allowing the system to choose, for each spatial view atom represented in the spatial view, the appropriate spatial representation for the scale manipulation (e.g displaying a network has different meanings at planning or civil engineering abstraction levels). The manipulation scale of a spatial view is provided if and only if at least one of its spatial view atoms is georeferenced.

Application of the spatial view to navigation knowledge context requires the specification of basic graph theory definitions:

Definition 1: A **graph** is a pair (N, E) where N is a node set and E a subset of the cartesian product $N \times N$. A node is defined by a type (Node_ type).

$N = \{n_1, ..., n_p\}$

$E = \{(n_i , n_j) / \ n_i \in N, n_j \in N \}$

Node_type = string

Definition 2: A **network** is a graph defined by a name (Name), and identified node and edge sets. A network models a communication system.

Network_type = [Name: string, Nodes: {Node_type}, Edges: {Edge_type}]
Edge_type = [Name: string, n_i: Node_type, n_j : Node_type]

The following example describes a spatial view which represents a displacement between an origin (e.g. EPFL_Entrance) and a destination (e.g. Arrival) through a network managed by a collection (e.g. Network_Graph). The representation of this displacement is built with a spatial view atom describing its layout (e.g. Track), a spatial view atom representing the network support (e.g. Network) and a spatial view atom situating the spatial environment (e.g. Buildings).

```
SVA: [  Name:                'Track'
        Collection:          {'Network_graph'}
        Spatial_Query:
             ( 'Select       Path ('EPFL_Entrance','Arrival')
               From          Network_graph'   )
        Visualization_Op:     {'bold'}            ]

SVA: [  Name:                'Network'
        Collection:          {'Network_Graph'}
        Spatial_Query:
             ( 'Select       Spatial_Rep
               From          Network_Graph'   )
        Visualization_Op:    { 'default' }        ]

SVA: [  Name:                'Buildings'
        Collection:          {'Buildings' }
        Spatial_Query:
             ( ' Select      Spatial_Rep
                 From        Buildings' )
        Visualization_Op:     {'hatch'}           ]
```

A spatial view example (EPFL_Campus) is created from these spatial view atoms (Figure 1). It specifies the respective roles of each spatial view atom (Order attribute). The manipulation scale of this spatial view is an interval centered on the scale of ″1/ 10,000″.

```
SV:  [  Name:           'EPFL_Campus',
        Order:          [      Essential:        {'Track'},
                               Important:        {'Network'},
                               Useful:           {'Buildings'}         ] ,
        Manipulation_Scale: { [ Min_Limit:       1/25, 000,
                               Scale_Reference:  1/10, 000,
                               Max_Limit:        1/5, 000    ]    }       ]
```

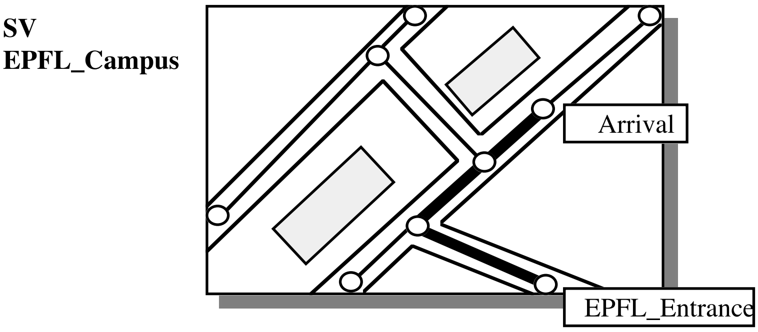

Figure 1 - Spatial view example

IV. Navigation knowledge model

The navigation knowledge model is built on a set of spaces represented by different abstraction levels and by semantic connections between these spaces: the spatial collages. Spatial collage semantics are specified by definition 3, and that of the navigation knowledge graph by definition 4.

Definition 3: A **spatial collage** describes a change of space in the description of a navigation process (e.g. from regional to local scale). Each represented space is defined by a distinct spatial view. A spatial collage is identified by a name (Name). The spatial collage translates an oriented semantic relationship between two spatial views identified by their names (SV_from, SV_to) (e.g. a spatial collage between a first spatial view representing a network at a regional scale, and a second spatial view representing the extension of this network at a local scale). The orientation represents the displacement direction.

Definition 4: A **navigation knowledge graph** is an application of the network concept. It is defined by a type (Process_Graph_type). The type specifies the name (Name), the node set (Nodes) and the edge set (Edges) of the navigation knowledge graph. Nodes model spatial views, edges model spatial collages. This graph represents the logical support of navigation knowledge.

Collage_type = [Name: string ,
SV_from: string ,
SV_to: string]

Process_Graph_type = [Name: string ,
Nodes: { Node_type } ,
Edges: { Collage_type }]

A navigation knowledge graph is defined using spatial views as nodes and spatial collages as edges. The hierarchical character of navigation knowledge leads to the definition of the incremental displacement element within the same spatial view (at the same abstraction level).

Definition 5 A **section** represents a displacement (without cycle) between an origin and a destination evaluated on a network within the same spatial view (at the same abstraction level). A section is defined by a type (Section_type) and as a network.

Section_type = Network_type

The semantic spatial collage associating two sections can be spatially materialized by ″connecting″ the relevant spatial instances of the two sections.

Definition 6: A **connection** is the spatial materialization of the spatial collage of two spatial views. Connection instances are identified by graph functions that give, for each of the two sections, the nodes which allow the connection (end of the section n, origin of the section n+1). There is only one connection for a given spatial collage. A connection is defined by its name (Name), the network names which define the sections (Section_ from_Name and Section_to_Name) and the nodes which allow the connection (Node_End and Node_Origin) by the graph functions (End) and (Origin).

End: Section_type --> Node_type
Origin: Section_type --> Node_type

End is defined by the n_i such as $\Gamma^+ (n_i) = \emptyset$; Origin is defined by the n_i such as $\Gamma^- (n_i) = \emptyset$ (Γ^+ and Γ^- are standard graph functions which identify respectively the nodes which have no successor and no predecessor).

Connection_type = [Name: string ,
Section_from_Name: string ,
Node_End: Node_type ,
Section_to_Name: string ,
Node_Origin: Node_ type]

The following example illustrates the concepts of spatial collage and connection (Figure 2). A navigation knowledge graph is defined between two spatial views (Highway and EPFL_Campus) by a spatial collage (EPFL_Arrival) of these two spatial views and by a connection (C2) between the corresponding sections (Highway_to_EPFL and Track) of these two spatial views (respectively Highway and EPFL_Campus). The Highway spatial view represents two spatial view atoms: a section (Highway_ to_EPFL) and a lake that gives a geographical description (Lake). The manipulation scale of the spatial view (Highway) is an interval centered on ″1/ 50,000″.

```
SV: [  Name:          ‘Highway’,
       Order:         [Essential:          {‘Highway_to_EPFL’} ,
                      Important:           { } ,
                      Useful:              {‘Lake’}   ]    ,
       Manipulation_Scale:
                      {  [  Min_Limit:     1/75,000  ,
                      Scale_Reference:     1/50,000  ,
                      Max_Limit:           1/25,000   ]         }   ]

Collage_type: [       Name:                ‘EPFL_Arrival’ ,
                      SV_from:             ‘Highway’,
                      SV_to:               ‘EPFL_Campus’   ]

Connection_type = [ Name:                  ‘C2’ ,
                      Section_from_Name:   ‘Highway_to_EPFL’,
                      Node_End:            ‘Highway_Exit’ ,
                      Section_to_Name:     ‘Track’ ,
                      Node_Origin:         ‘EPFL_Entrance’   ]
```

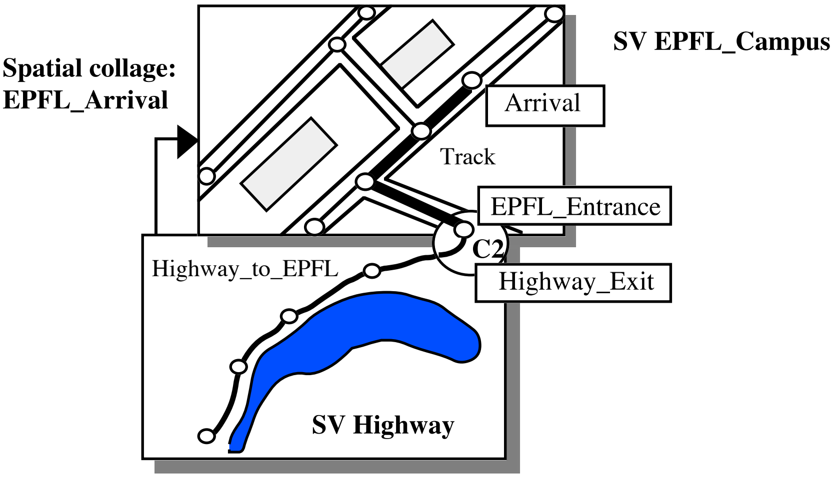

Figure 2 - Spatial collage and connection examples

A spatial collage of two spatial views does not require a connection if the represented navigation knowledge is defined at the general abstraction level. The spatial collage ensures the semantic association of sections; the connection ensures the spatial materialization of this association. A spatial collage is given by an intentional and declarative process controled by the user, while a connection is constrained by the continuity of the graph. The next definition qualifies connection cases.

Definition 7: A **convergent connection** (resp. **divergent connection**) is a connection whose the instances (Node_Origin and Node_End) which represent the nodes are spatially merged (resp. non merged) according to geometrical tolerances. Convergent connections ensure the spatial continuity (resp. the spatial discontinuity) of the navigation knowledge graph.

Divergent connections provide flexibility during a preliminary evaluation of a navigation knowledge graph (without a strict spatial continuity constraint, e.g. C2 connexion in the Figure 2). Divergent connections allow default reasoning (i.e. assumption that there is a connection solution whose details will be later identified by the user). A connection can associate two spatial collection instances defined by different spatial representation types but corresponding to the same spatial entity in the real world (e.g. a highway section defined by an area spatial data type in a first spatial view and a highway section defined by a line spatial data type in a second spatial view). A connection can associate two spatial entities situated on non peripheral spatial view positions (semantic continuity but non visual continuity between sections). The connection characterization may be extended, through abstract data types, by the integration of perceived elements such as photographic or symbolic images describing the connection place. These information elements can constitute visual landmarks for a better understanding of a navigation knowledge representation. We introduce the route notion to represent a navigation knowledge displacement:

Definition 8: A **route** represents the ordered vision of a navigation knowledge graph. A route describes the mental representation of a displacement through multidimensional spaces represented by spatial views and associated by spatial collages. A route is defined by a name (Name) and its displacement representation (Navigation_type). This displacement is represented by a spatial view followed by a list of spatial collage - spatial view pairs.

Navigation_type = [SV_Name: Node_type ,
Sequence: ([Collage_Name: string ,
SV_Name: Node_type])]

Route_type = [Name: string,
Trace: Navigation_type]

The construction of a route is a function (Trace) of the navigation knowledge graph:

Trace: Process_Graph_type --> Route_type

V. Navigation knowledge manipulation

V.1. Definition Language

The components of the navigation knowledge model are specified using a classic definition language allowing basic creation and deletion operations. Spatial views and spatial collages allow the design of a navigation knowledge graph (Process_Graph_type). The realization of a spatial view is an explorative process in which the user proceeds by iterations before identifying the manipulation scale which corresponds to the right abstraction level and a coherent order of the spatial view atoms. The route is built according to each user objective and by the definition of a graph order (Route_type). The displacement will be defined by sections specifying its layout in each spatial view and by connections between these sections if allowed by the corresponding abstraction levels. The creation operation set takes the general form of an operator create <component structure of the component>. Conversely, deletion operations take the general form delete <component name>.

V.2. Manipulation Language - external level

The data manipulation language relies on classical mechanisms to handle interactions with Process_Graph_type. Two general levels are defined: an identification process and a manipulation (selection and visualisation) process.

V.2.a *Identification operators*

Identification operators allow the extraction of a spatial view (SV_Extract) from its name, and extraction of the origin spatial view (SV_Origin) and the last spatial view (SV_End) of a navigation knowledge graph (Process_Graph_type). A SV_Next operator gives the following spatial view from a spatial view of a navigation knowledge graph. Similarly, an operator identifies a spatial collage (Collage_Extract) member of a navigation knowledge graph (Process_Graph_type).

SV_Extract: Process_Graph_type x string --> SV_type
SV_Origin: Process_Graph_type --> SV_type
SV_End: Process_Graph_type --> SV_type
SV_Next: Process_Graph_type x SV_type --> SV_type

SV_Origin defines the spatial view modeled by n_i graph node, such as $\Gamma^{-}(n_i) = \emptyset$

SV_End defines the spatial view modeled by the n_i graph node, such as $\Gamma^{+}(n_i) = \emptyset$

Collage_Extract: Process_Graph_type x string --> Collage_type

V.2.b *Selection and visualization operators*

An operator allows selection of a route by its name (Route_Select_by_Name). Let Database_type denote the type modelling the database.

Route_Select_by_Name:	string x Database_type	-->	Route_type

Operators are provided to visualize a spatial collage (Collage_Display), a spatial view (SV_Display) and a route (Route_Display). Displayed elements are defined by types (respectively Collage_Display_type, SV_Display_type and Route_Display_type).

Collage_Display:	Collage_type	-->	Collage_Display_type
SV_Display:	SV_type	-->	SV_Display_type
Route_Display:	Route_type	-->	Route_Display_type

V.3. Manipulation language - internal level

A route can be expressed according to several abstraction levels. At each abstraction level there is a specific sequence of spatial views and spatial collages represented by a Process_Graph_type. A spatial view can be decomposed into several spatial views connected by spatial collages; this operation is realized by the application of a development operator (Develop) on a Process_Graph_type. Conversely, a spatial view sequence connected by spatial collages can be grouped within a spatial view; this operation is realized by the application of a group operator (Undevelop) to a Process_Graph_type. These operators allow changes in the abstraction levels of a route representation [Mai 95].

V.3.a *Develop Operator*

The signature of the Develop operator is defined as:

Process_Graph_type x Process_Graph_type x Node_type -> Process_Graph_type

The first Process_Graph_type is the graph which models the initial route. The second Process_Graph_type is the graph which specifies a more detailed part of the initial route. Node_type is the node of the graph to be developed (the spatial view whose level of detail will be increased). This node can be obtained, for example, by using the name of the spatial view obtained by a query on a Route_Display_type. The result is a Process_Graph_type (a graph which models a route with a greater level of detail than the initial route).

This operator specification is based on graph properties modeled by Process_Graph_type. These graphs represent routes (without cycles). Therefore, they have an origin and an extremity. The application of the Develop operator leads to the study of three cases: (a) the developed node is the origin node; (b) the developed node is the extremity; (c) the developed node is neither the origin, nor the extremity (nodes model spatial views, edges model spatial collages).

Therefore, the specification of the Develop operator may be identified by a graph expression:

$G_1 (N_1, E_1) \times G_2 (N_2,E_2) \times N \rightarrow G_3 (N_3, E_3)$.

S_1 and S_2 are the sections modeled by the graphs G_1 and G_2.

Case (a) specifications are the following for the development of the node n_d (Figure 3):

e_{out} is defined as: $e_{out} \in E_1 / \exists\ n_{out} \in N_1 / e_{out} : (n_d, n_{out})$

$N_3 = N_1 - \{n_d\} \cup N_2$

$E_3 = E_1 - \{e_{out}\} \cup E_2 \cup \{ (End (S_2), n_{out}) \}$

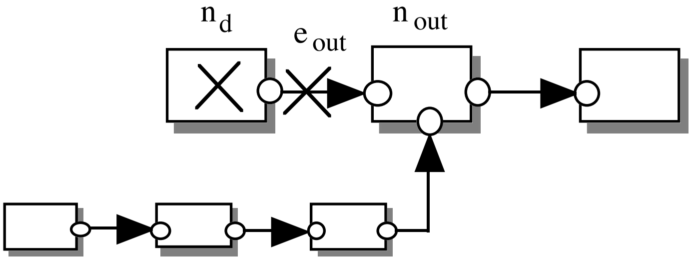

Figure 3 - Origin node graph Develop

Case (b) specifications are the following for the development of the node n_d (Figure 4):

e_{in} is defined as $e_{in} \in E_1 \,/\, \exists\, n_{in} \in N_1 \,/\, e_{in} : (n_{in}, n_d)$
$N_3 = N_1 - \{n_d\} \cup N_2$
$E_3 = E_1 - \{e_{in}\} \cup E_2 \cup \{ (n_{in}, \text{Origin } (S_2))\}$

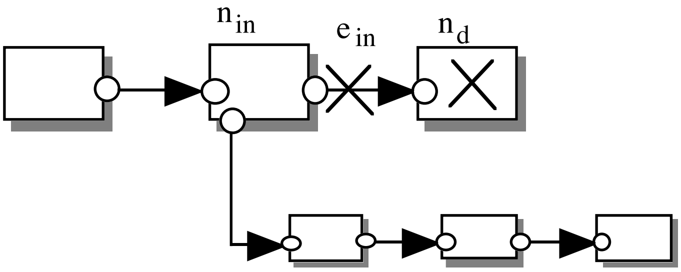

Figure 4 - Last node graph Develop

Case (c) specifications are the following for the development of the node n_d (Figure 5):

$e_{in} = e_{in} \in E_1 \,/\, \exists\, n_{in} \in N_1 \,/\, e_{in} : (n_{in}, n_d)$
$e_{out} = e_{out} \in E_1 \,/\, \exists\, n_{out} \in N_1 \,/\, e_{out} : (n_d, n_{out})$
$N_3 = N_1 - \{n_d\} \cup N_2$
$E_3 = E_1 - \{e_{in}, e_{out}\} \cup E_2 \cup \{ (n_{in}, \text{ Origin } (S_2)), (\text{End}(S_2), \; n_{out})\}$

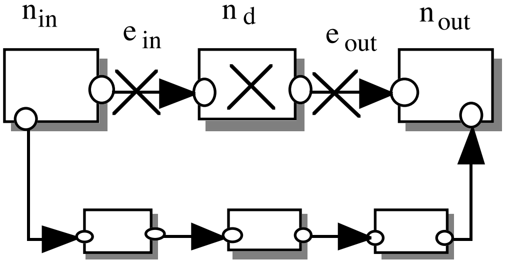

Figure 5 - Standard node graph Develop

V.3.b *Undevelop Operator*

The signature of the Undevelop operator is defined as:

Process_Graph_type x Process_Graph_type x Node_type -> Process_Graph_type

The first Process_Graph_type is the graph which models the initial route. The second Process_Graph_type is the graph which models the route to be simplified (i.e. a sub-route of the initial route - a graph connected to and resulting from a Develop operator application). Node_type is the node of the graph that represents the abstraction of the deleted sub-route of the initial graph (i.e., the spatial view representing the sub-route). The result is a Process_Graph_type (a graph which models a route with a less precise level of detail than the initial route).

These operator specifications are also derived from graph properties modeled by a Process_Graph_type. As for the Develop operator, these graphs model routes (without cycles). Therefore, they have an origin, and an extremity. There are three cases when applying the Undevelop operator application: (a) the grouping node is placed at the origin node; (b) the grouping node is placed at the extremity node; (c) the grouping node is placed neither at the origin, nor at the extremity. Therefore, the specification of the Undevelop operator is identified by a graph expression:

$G_1 (N_1, E_1) \times G_2 (N_2,E_2) \times N \to G_3 (N_3, E_3)$.

S_1 and S_2 are the sections modeled by the G_1 and G_2 graphs.

Case (a) specifications are the following for the grouping of the node n_u (Figure 6):

e_{out} is defined as: $e_{out} \in E_1 / \exists\, n_{out} \in N_1 / e_{out}$: (End (S_2), n_{out})

$N_3 = N_1 \cup \{n_u\} - N_2$

$E_3 = E_1 - \{e_{out}\} - E_2 \cup \{ (n_u, n_{out}) \}$

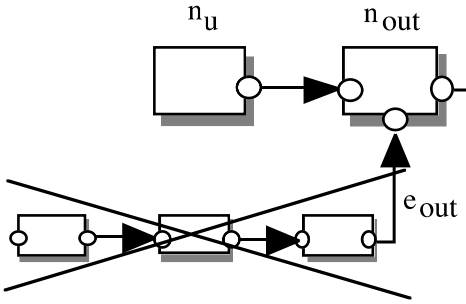

Figure 6 - Origin node graph Undevelop

Case (b) specifications are the following for the grouping of the node n_u (Figure 7):

e_{in} is defined as $\in E_1 / \exists\, n_{in} \in N_1 / e_{in}$: (n_{in}, Origin(S_2))

$N_3 = N_1 \cup \{n_u\} - N_2$

$E_3 = E_1 - \{e_{in}\} - E_2 \cup \{ (n_{in}, n_u)\}$

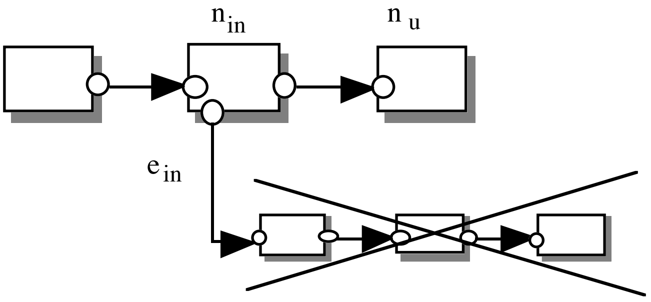

Figure 7 - Last node graph Undevelop

Case (c) specifications are the following for the grouping of the node n_u (Figure 8):

$e_{in} = e_{in} \in E_1 / \exists\, n_{in} \in N_1 / e_{in}$: (n_{in}, Origin (S_2))

$e_{out} = e_{out} \in E_1 / \exists\, n_{out} \in N_1 / e_{out}$: (End (S_2), n_{out})

$N_3 = N_1 \cup \{n_u\} - N_2$

$E_3 = E_1 - \{e_{in}, e_{out}\} - E_2 \cup \{ (n_{in}, n_u), (n_u, n_{out})\}$

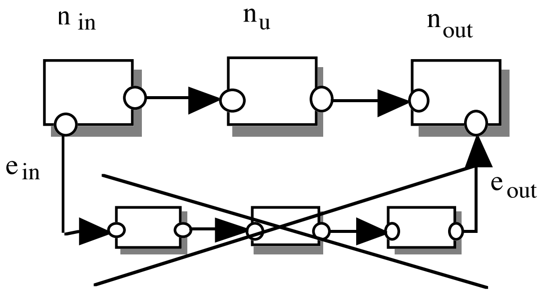

Figure 8 - Standard node graph Undevelop

V.3.c Example

We introduce a new spatial view, Office, that gives a symbolic representation of the arrival place for this route example, and a new spatial view, EPFL, that gives a symbolic representation. The next figure illustrates the application of the Develop operator applied to the EPFL spatial view (Figure 9).

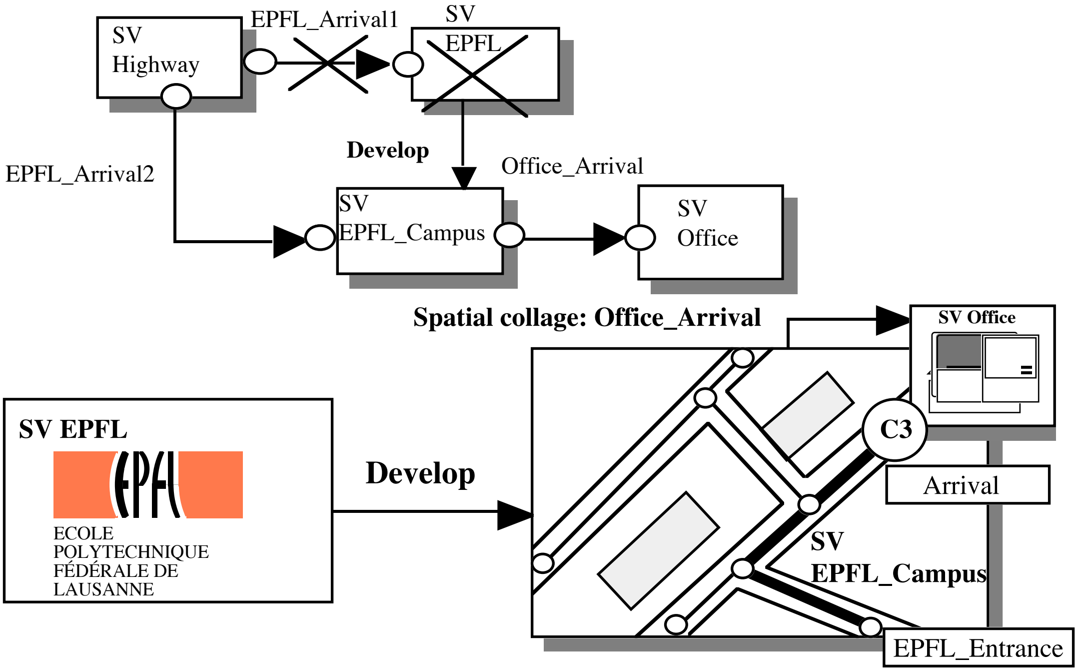

Figure 9 - Develop operator application

The graph concept ensures route continuity during the application of these operators. It allows abstraction changes in the representation of the route and the use of corresponding spatial views for each of these abstraction levels. The model allows representation of partial knowledge within the route description (e.g. a user knows that a route leaves an airport but is unable to describe the airport site displacement).

The route scale is the union of the manipulation scales of the spatial views which compose this route. Considering the diversity of spatial abstractions used in a navigation process, this scale notion gives a global vision of the set of abstraction levels.

The ordered representation of a route allows the application of a textual operator (Textual) that gives a descriptive textual expression using the names of corresponding spatial views. The textual form is provided by an expression From <Initial view name> followed by a sequence of Path to <Name of the next view> applied to the spatial views part of the Sequence attribute of the Navigation_type. The signature of this textual operator is given by the following expression:

Textual: Route_type --> string

VI. Application

A navigation process example (Geneva_Airport_to_EPFL) begins at the airport (Airport_Site), and continues through a highway (Highway) to the EPFL. At a general abstraction level, a first example of route is:

```
Route:
    [ Name:        'Geneva_Airport_to_EPFL',
      Trace:       [   SV_Name:    'Airport_site',
                       Sequence:   ( [  Collage_Name:  'Airport_exit' ,
                                        SV_Name:        'Highway'   ] ,
                                     [  Collage_Name:  'EPFL_Arrival1' ,
                                        SV_Name:        'EPFL'  ]    )     ] ]
```

The Airport_Site and EPFL spatial views provide significant symbols at this abstraction level. Application of the Textual operator documents the visualization represented by the textual form “From Airport_site Path to Highway Path to EPFL”. Figure 10 presents a visualization example of this route (the display arrangement of the spatial views representing the route is an interface level task).

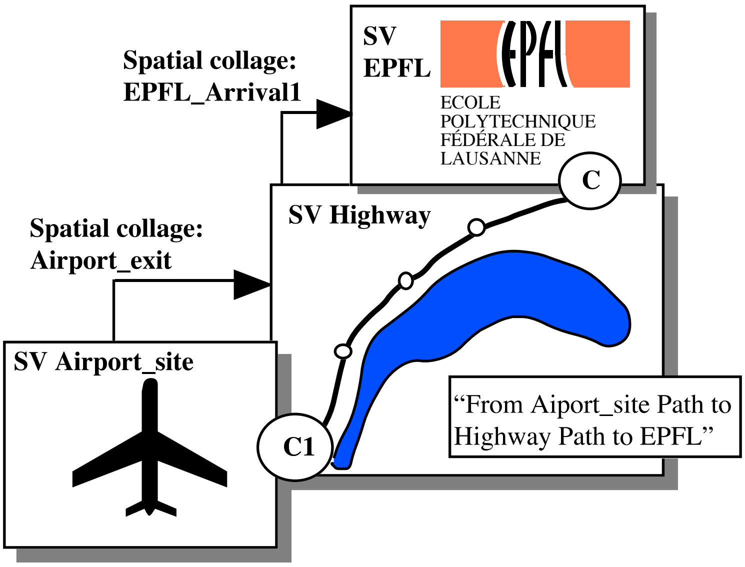

Figure 10 - Route example at a general abstraction level

A second route example illustrates a navigation process at a more precise abstraction level. The objective of this second route example is to specify the presentation of the spatial views which compose this navigation process. The previous EPFL symbolic spatial view is decomposed by the application of a Develop operator applied to the EPFL spatial view.

```
Route:
    [ Name:        'Geneva_Airport_to_EPFL',
      Trace:       [   SV_Name:    'Airport_site',
                       Sequence:   ( [  Collage_Name:  'Airport_exit' ,
                                        SV_Name:        'Highway'   ] ,
                                     [  Collage_Name:  'EPFL_Arrival2' ,
                                        SV_Name:        'EPFL_Campus' ] ,
                                     [  Collage_Name:  'Office_Arrival' ,
                                        SV_Name:        'Office'   ]    )     ]     ]
```

The spatial view Office gives a visual signal of the destination place. The textual description is more complete than that of the previous example. Figure 11 illustrates this second route representation example (C1 and C3 are divergent connection examples).

These route representations may be extended by various user experiences or knowledge. The route sequence allows initialization of condition-action events (condition = spatial collage, action = displacement initialisation within a spatial view) necessary to generate a navigation process [Kui 78]. A spatial collage gives a passage condition for continuing navigation through a section. Actions to undertake (e.g. right, left, until, next, [Maa 93]) for the route displacement and associated positional operators [Kui 78] can make use of this model.

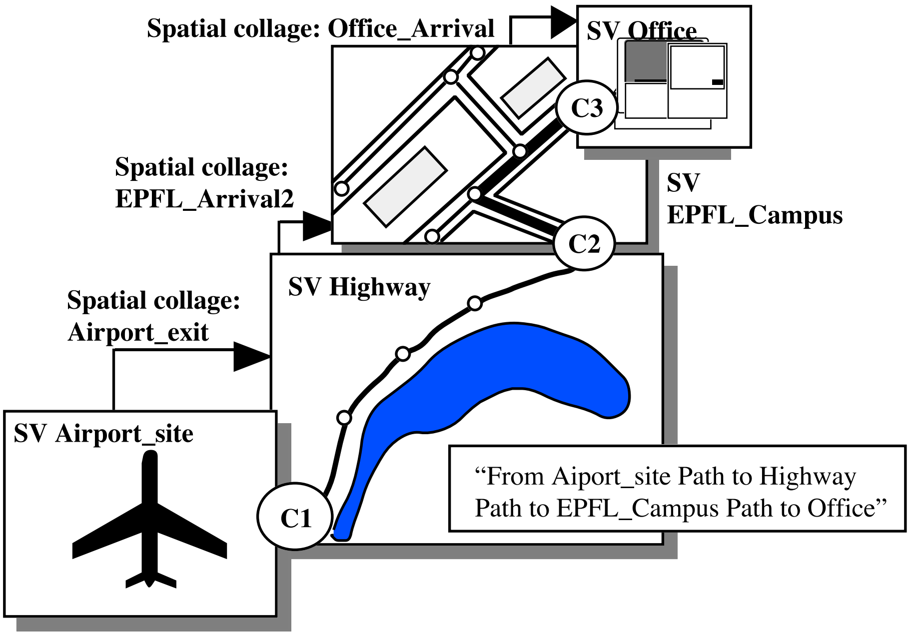

Figure 11 - Route example at a more precise abstraction level

VII. Conclusion

The coordinate route representation with multidimensional associated spaces meets the needs of navigation process users. It preserves displacement memory and facilitates its assimilation, favours reusability and ensures updating.

The proposed model allows a displacement action to be situated within its geographical context through complementary abstraction levels that accept partial knowledge. The spatial view gives a representation framework for navigation knowledge. It associates the visualization of a route with multidimensional spaces that allow it to be situated, including significant visual landmarks and textual descriptions. Continuity of route representation is ensured by the graph concept applied to spatial views and spatial collages. Spatial collages are spatially materialized by connections. A route trace in each spatial view spaces is described by a section. Develop and Undevelop operators allow abstraction level changes within the route representation.

This proposal may be applicable to network guidance applications, tourism or professional displacement evaluations. Nautical or aerial navigation are other potentially suitable applications. Further work involves the definition of a manipulation language and the realisation of a demonstrative prototype.

References

[Adi 87] Adiba, M. E., Modeling complex objects for multimedia databases. *Entity-Relationship Approach: Ten Years of Experience in Information Modelling*, Spaccapietra, S. eds., North-Holland, Dijon, France, November 17-19, 1987, pp. 89-117.

[Bur 86] Burrough, P. A., *Principles of Geographical Information Systems for Land Resources Assessment*, Oxford: Clarendon Press, United-Kingdom, 1986.

[Che 89] Chen, S., A spherical model for navigation and spatial reasoning. *Mapping and Spatial Modelling for Navigation*, Pau, L. F. ed., Springer-Verlag, Fano, Denmark, August 21-25, 1989, pp. 59-72.

[Cla 94] Claramunt, C. and Mainguenaud, M., Identification of a definition formalism for a spatial view. *Advanced Geographic Data Modelling*, Molenaar, M. and de Hoop, S. eds., Netherlands Geodetic Commission, Delft, The Netherlands, September 12-14, 1994, pp. 191-203.

[Cla 95] Claramunt, C. and Mainguenaud, M., Dynamic and flexible vision of a spatial database. *Proceedings of the DEXA'95 Int. Workshop on Database and Expert System Applications*, Revell, N. and Min Tjoa, A. eds., Omnipress, London, United-Kingdom, September 4-5, 1995, pp. 483-492.

[Gol 95] Golledge, R. G., Path selection and route preference in human navigation: A progress report. *Spatial Information Theory: A Theoretical Basis for GIS*, Frank, A. U. and Kuhn, W. eds., Springer-Verlag, Semmering, Austria, September 21-23, 1995, pp. 207-222.

[Ken 80] Kennington, J. L. and Helgason, R. V., *Algorithms for Network Programming*, Wiley, J. ed., New-York, 1980.

[Kui 78] Kuiper, B., Modeling spatial knowledge. *Cognitive Science*, Vol. 2, 1978, pp. 129-153.

[Lyn 60] Lynch, K., *The Image of the City*, MIT Press, Massachusetts, 1960, USA.

[Maa 93] Maaß, W., A Cognitive model for the process of multimodal, incremental route descriptions. *Theories and Methods of Spatio-temporal Reasoning in Geographic Space*, Frank, A. U., Campari, I. and Formentini, U. eds, Springer-Verlag, Elbe Island, Italia, September 19-22, 1993, pp. 1-13.

[Mai 95] Mainguenaud, M., The modelling of the geographic information system network component. *International Journal of Geographical Information Systems,* v. 9 n. 6, Taylor and Francis, 1995, pp. 575-593.

[Mar 91] Mark, D. M. and Frank, A. U., *Cognitive and Linguistic Aspects of Geographic Space*, Kluwer Academic Publishers, NATO ASI series, 1991.

[Pau 89] Pau, L. F., *Survey, Mapping and Spatial Modelling for Navigation*, Pau, L. F. ed., Springer-Verlag, Fano, Denmark, August 21-25, 1989, pp. 1-9.

[Peu 88] Peuquet, D. J., Representations of geographic space: Toward a conceptual synthesis. *Annals of the Association of American Geographers*, 78 (3), 1988, pp. 375-394.

[Ren 95] Rennison, E. and Strausfeld, L., The Millennium project. *Spatial Information Theory: A Theoretical Basis for GIS*, Frank, A. U. and Kuhn, W. eds., Springer-Verlag, Semmering, Austria, September 21-23, 1995, pp. 69-91.

[Sto 93] Stonebraker, M., Chen, J., Nathan, N., Paxson, C. and Wu, J., Tioga: Providing data management support for scientific visualization applications. *Proceedings of the Very Large Data Base Conference*, Dublin, Ireland, August 24-27, 1993.

[Tve 93] Tversky, B., Cognitive maps, cognitive collages, and spatial mental models. *Theories and Methods of Spatio-temporal Reasoning in Geographic Space*, Frank, A. U., Campari, I. and Formentini, U. eds., Springer-Verlag, Elbe Island, Italia, September 19-22, 1993, pp. 14-24.

[Tho 93] Thorndyke, P. W. and Hayes-Roth, B., *Differences in Spatial Knowledge Acquired from Maps and Navigation*, Office of Naval Research, Santa Monica, USA, 1993.

[Tim 92] Timpf, S., Volta, G. S., Pollock, D. W. and Egenhofer, M. J., A conceptual model of wayfinding using multiple levels of abstraction. *Theories and Methods of Spatio-temporal Reasoning in Geographic Space*, Frank, A. U., Campari, I. and Formentini, U. eds, Springer-Verlag, Pisa, Italia, September 1992, pp. 348-362.

[Woo 95] Woodruff, A. et al., Zooming and tunneling in Tioga: Supporting navigation in multidimensional space. *Proceedings of Visual Database Systems Conference*, Spaccapietra, S. and Jain, R. eds., Chapman et Hall, Lausanne, Switzerland, March 27-29, 1995, pp. 323-332.