IMPLEMENTING SEMANTIC REFERENCE SYSTEMS

Werner Kuhn and Martin Raubal

Institute for Geoinformatics, University of Münster,
Robert-Koch-Str. 26-28, 48149 Münster, Germany.
{kuhn|raubal}@ifgi.uni-muenster.de

Abstract

The analogy of Semantic Reference Systems proposed in (Kuhn in press) is being explored here with respect to the computational mechanisms it suggests. Semantic referencing, grounding in a semantic datum, semantic projection, and semantic transformation are defined and demonstrated through an implementation of a semantic reference system for a simple vehicle navigation model. The idea of wrapping legacy data or services is shown to correspond to semantic referencing, which suggests a straightforward procedure to derive wrappers from semantic reference systems.

1. INTRODUCTION

How can we progress from today's informal models of semantics to a state where the meaning of geographic information can be accessed, read and manipulated by systems as well as humans? Four hundred years ago, geometric information was similarly unstructured and informal as semantic information is today. Then, René Descartes invented coordinate systems, laying the groundwork for what we now call spatial reference systems. These reference systems enable GIS users to share spatially referenced information across multiple system platforms and integrate it with respect to their geometry, but not their semantics.

A corresponding formal foundation is required for the semantics of geo-referenced information. It is a precondition to achieve *semantic interoperability*, i.e., the ability of systems and services to interoperate through a common understanding of geographic information. Since GIS are often defined as holding two kinds of data, geometric and thematic (Longley et al. 2001), and since the reference systems for geometric data are now well understood and standardized (ISO 2002c), it appears natural to engage in the design and implementation of *semantic* reference systems.

The parts and processes of spatial reference systems suggest direct analogues for the case of semantics. This paper explores this analogy, introducing the notions of a semantic reference system as a whole, of a semantic datum and reference frame as its parts, and of semantic referencing, projection, and transformation as its processes. The purpose is to establish a better understanding of parts and processes in the design and use of ontologies. Introducing new terminology is not our goal, but rather a way to shed light on ongoing work through a metaphor rooted in the well-understood domain of spatial reference systems.

Section 2 describes the main ideas behind semantic reference systems. It further introduces Haskell, the functional language used in this paper, and our running example from the domain of vehicle navigation systems. In section 3 the core elements of a semantic reference system, i.e., the semantic datum and the semantic reference frame, are specified. Section 4 defines the processes of semantic reference, semantic projection, and semantic transformation in analogy to their spatial counterparts. Section 5 presents conclusions and suggests some directions for future research.

2. SEMANTIC REFERENCE SYSTEMS

A theory of spatial reference systems establishes precise notions of a geodetic datum, coordinate systems, map projections, and coordinate transformations (ISO 2002c). Similarly, temporal data refer unambiguously to temporal reference systems (ISO 2002b), such as calendars, and can be transformed from one to another. Spatial and temporal reference systems allow us to position entities dynamically relative to other entities in the real world, to transform their GIS representations from one system to another, and to integrate spatial and temporal references across multiple systems. A theory of *semantic* reference systems will similarly enable producers and users of geographic information to explain the meaning of thematic data, to translate this meaning from one information community to another, and to integrate data across heterogeneous semantics.

Semantic reference systems go beyond ontologies, but ontologies constitute a core component of them. According to a common definition, an ontology specifies the conceptualization of the world to which the data in an information system refer (Gruber 2002). In other words, it provides a frame of reference for the vocabulary used in the system. For example, an ontology might specify what the term 'forest' means in a vegetation data model. This referencing facility turns out to be a core function of semantic reference systems, but not the only one. They should also support projecting data models to simpler ones and translating data from one model to another.

Today's practice of using ontologies does not meet these goals, neither in the field of geographic information, nor beyond. Domain ontologies are often constructed haphazardly, without explicit and formal ties to upper-level ontologies. The referencing of data to ontologies can consist in a simple URL indicating where users can find definitions of terms. These definitions are sometimes stated in a natural or semi-formal language, allowing for different interpretations. If a data set uses only a sub-set of the terms in a data model (which is the normal case), it is generally not possible to project the ontologies to their corresponding parts, while retaining all necessary definitions.

Finally, and most importantly, the process of semantic translation, proposed in (OGC 1998), has not yet received a theoretical foundation with clearly defined operations that users can apply to data and ontologies. Most existing implementations of semantic translators are hand-coded mappings between informally specified data models. They cannot be generalized or easily adapted to changing requirements, and they do not provide us with an understanding of the translation process. For all these reasons, we believe that the current patchwork approach to ontologies should be evolved into an integrating theory of semantic reference systems.

2.1 The elements of semantic reference systems

The analogy between spatial and semantic reference systems suggests structural and functional requirements. Certain structures need to be defined as inputs to and results of semantic processes. This paper introduces two key structural notions: the idea of a *semantic datum*, grounding the meaning of basic terms outside the system; and the idea of a *semantic reference frame*, acting like a coordinate system as a formally defined framework to which terms can be related to obtain meaning.

On the functional side, processes need to be defined that support the explanation and transformation of semantics in geographic data. We first define *semantic referencing* as the basic process for relating the terms in a data model to a semantic reference frame. Then, in analogy to map projections, we introduce a process of *semantic projection*, supporting mappings from more complex to simpler data models. Finally, we give a precise characterization for the process of *semantic translation* as a mathematical transformation from one semantic reference system to another.

Spatial reference systems can be seen as a special case of semantic reference systems: they explain the meaning of coordinates. This restriction to the difficult general semantics problem has allowed for highly successful solutions to the spatial case. The best strategy for progress towards general semantic reference systems is, thus, to find the next easiest special cases to solve. For example, one could start with gazetteers, since they map the semantic problems posed by location names (e.g., recognizing synonyms or other relationships) to the corresponding spatial problems. Our running example addresses the slightly more challenging problem of attaching semantics to a graph data model, in particular to the edges in a simple navigation model.

2.2 The tool and the running example

We apply Hugs, a dialect of the Haskell functional language standard (Hudak 2000), to formalize, implement, and test a semantic reference system for a navigation model. In a nutshell, the necessary syntax is the following[1]:

- *algebraic data types:* `data Car = Car Name Position`
  (introducing user-defined types with a constructor function of the same name),
- *type synonyms:* `type Position = [Int], type Name = String`
  (calling pre-defined types differently),
- *type classes:* `class Named n where name :: n -> Name`
  (collecting types sharing some behavior),
- *instances:* `instance Named Car where name (Car n p) = n`
  (inheriting the class behavior to a data type)
- *contexts:* `instance Named object => Eq object where …`
  (asserting a constraint on the types in a class)
- *type dependencies:* `… | link -> object`
  (stating that the type of one class parameter is determined by another)

For details about the implementation language, readers are referred to (Hudak 2000) and to the web site indicated in the footnote.

Let us assume that a data model for vehicle navigation systems introduces nodes (with identifiers) and edges connecting nodes:

```
data Node = Node Name
data Edge = Edge Node Node
```

Road elements and ferry connections are defined based on edges:

```
data RoadElement = RoadElement Edge
data FerryConnection = FerryConnection Edge
```

[1] The Hugs code for this paper is available from the *Experiments* section of http://musil.uni-muenster.de. Hugs interpreters can be downloaded freely from http://www.haskell.org.

This is a simplified specification of the Geographic Data Files (GDF) data model, levels 0 and 1 (ISO 2001). Additionally, we define the implied application concepts: cars and car ferries, both located at nodes:

```
data Car = Car Node
data CarFerry = CarFerry [Car] Node
```

This completes the definition of the elements for our domain ontology. The necessary axioms will be supplied in section 4, as part of the semantic referencing.

3. SEMANTIC DATUM AND REFERENCE FRAME

In this section, the core elements of a semantic reference system are introduced: the *semantic datum*, which grounds terms in an understanding outside the reference system, and the *semantic reference frame*, acting as an upper-level ontology to which data models commit through their domain ontologies.

3.1 Semantic Datum

A datum in a spatial reference system defines the position of the origin, the scale, and the orientation of the axes of a coordinate system. Typical examples are a geodetic or a vertical datum. A datum anchors the geometry model to the real world, by defining its relative position with respect to the physical earth.

Correspondingly, we define a *semantic datum* as a mechanism for grounding the terms of a semantic reference system. By *grounding*, we mean the process of providing an interpretation for some terms outside a formal system, rather than just pointing to other terms within the system. Typically, the terms to be grounded are the most primitive ones in a given data model. The grounding can be a (non-circular) pointer to other ontologies, or an anchoring in the formally defined semantics of a programming language.

For our example of navigation system semantics, we adopt the semantics of Hugs as the datum. Implicitly, this occurs through the implementation of the semantic reference system in Hugs. All the Hugs syntax used in the reference system gets its meaning from the semantics of Hugs. In addition, a single line of Hugs code grounds the semantics of the data model's names (identifiers) explicitly in Hugs Strings. All other types are explained using this `Name` type and the Hugs language constructs (e.g., `data`):

```
type Name = String
```

Typically, the semantics of the implementation language for a reference system should only concern few and undisputed points in the referencing process, such as the (irrelevant) execution semantics of the *list* of cars on the ferry.

3.2 Semantic Reference Frame

In a spatial reference system, coordinate systems serve as mathematically well-defined frameworks to which positions can refer. For example, the precisely specified WGS 84 ellipsoid is used as the reference surface and coordinate system in the World Geodetic System of 1984 (Leick 1995). Fig. 1 shows an example of measuring the position of a point P on the geoid (an equipotential surface, which on average coincides with the global ocean surface) in terms of a reference ellipsoid, i.e., projection onto the ellipsoid's surface along its normal.

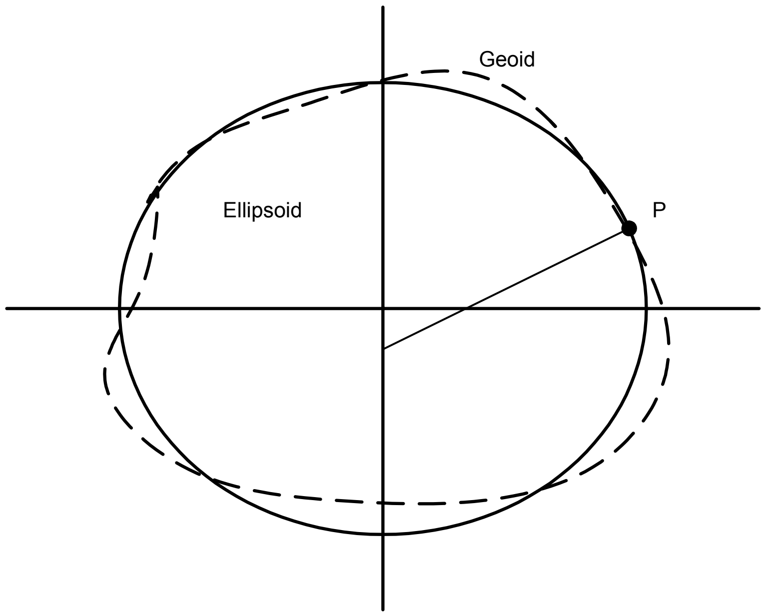

Fig. 1 Measuring the position of point P in terms of a reference ellipsoid.

Accordingly, we define a *semantic reference frame* as a formally defined, application-independent conceptual structure, called an upper-level or top-level ontology in the literature (Guarino 1998). As in the spatial case, this reference frame is the heart of a semantic reference system. It defines the conceptualization underlying the use of terms in a data model, and it extends beyond the single model to a generic structure that can be used for semantic referencing in multiple application domains and information communities.

In our example reference frame, we will first introduce some basic concepts: naming and location,

```
class Named object where name :: object -> name
instance Named object => Eq object where
        object1 == object2 = (name object1) == (name object2)

class LocatedAt object where location :: object -> Location
```

and then three generic concepts or 'schemata': links, paths, and surfaces. The Link schema introduces operations to ask for each end of the link and, given one end, to retrieve the other:

```
class Eq object => Link link object | link -> object where
        from, to :: link -> object
        other :: link -> object -> object
        other link object   | object == from link = to link
                            | object == to link = from link
                            | otherwise = error "not linked"
```

The Path schema adds the behavior of moving along to the Link schema:

```
class (Link path location, LocatedAt object) =>
  Path path location object | path -> location, path -> object where
        move :: object -> path -> object
```

The Surface schema offers the behavior we commonly associate with support provided by surfaces:

```
class Eq object => Surface surface object where
        putOn  :: object -> surface -> surface
        takeoff :: object -> surface -> surface
        isOn :: object -> surface -> Bool
```

Incidentally, these classes correspond to (and were inspired by) basic cognitive structures called image schemata (Johnson 1987), but this connection is not further explored here.

4. SEMANTIC REFERENCE, PROJECTION, AND TRANSFORMATION

Having established the notions of semantic datum and reference frame, it is now possible to define useful operations on them. In particular, the three processes of referencing data models to a reference system, projecting one data model to another, and translating semantics between data models are introduced.

4.1 Semantic Reference

Spatial reference systems serve to interpret coordinates. A coordinate value in a geospatial data model has to refer to a particular spatial reference system to be shareable (and interpretable) across different GIS (ISO 2002a).

Similarly, we propose that the types standing for feature names, attribute values, relationships, and operations in geospatial data models need to be given a reference to a semantic reference system. This process of semantic referencing is essential to any solution of a semantic problem. It establishes interpretation rules for the terms used in data models. Note that the reference points not to entities in the real world, but to conceptual structures that we have termed semantic reference frames above.

To better understand the crucial role and the particular form of this referencing process, consider again our example. Semantic references for the terms used in the navigation model are provided in the form of axioms, relating the terms used in the data model to the conceptual schemas introduced in the reference frame. We give here the complete set of axioms necessary to tie the navigation model to the upper-level ontology or reference frame.

First, we state that Nodes have names:

```
instance Named Node Name where
  name (Node n) = n
```

Cars and car ferries are located at nodes:

```
instance LocatedAt Car Node where
  location (Car node) = node
```

```
instance LocatedAt CarFerry Node where
  location (CarFerry cars node) = node
```

Then, we state that edges behave like Links:

```
instance Link Edge Node where
  from (Edge node1 node2) = node1
  to (Edge node1 node2) = node2
```

Next, road elements are Links as well, and Paths for cars:

```
instance Link RoadElement Node where
  from (RoadElement edge) = from edge
  to (RoadElement edge) = to edge

instance Path RoadElement Node Car where
  move (Car node) (RoadElement edge) = Car (other edge node)
```

The same goes for ferry connections, but they are Paths for car ferries, not cars:

```
instance Link FerryConnection Node where
  from (FerryConnection edge) = from edge
  to(FerryConnection edge) = to edge

instance Path FerryConnection Node CarFerry where
  move (CarFerry cars node) (FerryConnection edge)
   = CarFerry (map (Car.(other edge).location) cars) (other edge node)
```

Finally, we state that car ferries behave like Surfaces for cars:

```
instance Surface CarFerry Car where
  putOn aCar (CarFerry objects node) =
         if (node == location aCar)
         then CarFerry (aCar:objects) node
         else error "Car is not at CarFerry location"
  takeOff aCar (CarFerry objects node)
   = CarFerry (delete aCar objects) node
  isOn aCar (CarFerry objects node) = elem aCar objects
```

From a practical point of view, semantic reference boils down to *wrapping* information resources. Wrapping legacy data or information sources means to define access operations for them. The result is an interface specification and implementation, also known as an API (application programming interface). A wrapper for information sources accepts queries conforming to this interface, converts them into operations executable with the underlying information source, and transforms the obtained results into the API output parameters understood by an application.

Accordingly, the axioms given above relate the data types of the navigation application to the behavior specified in the reference frame. They completely specify, in a testable form, the necessary wrapper functions to access navigation data conforming to the specified data model (i.e., GDF).

4.2 Semantic Projection

In a *spatial* reference system, coordinates can be converted from one system to another. The datum to which they refer remains the same. The purpose of a coordinate conversion is usually to simplify computations, the simplest case being a shift of origins. The prototypical cases of conversions are map projections, where three-dimensional coordinates are projected onto a plane.

This dimensional reduction of coordinate projections provides the analogy for semantic reference systems. Many semantic problems involve a projection from one semantic space to another, where the second space is simpler or more abstract, in the sense of having less dimensions.

In our example, let us assume that the original data model is further simplified into one that contains only nodes and edges, without distinguishing road elements and ferry connections. This semantic projection of the data model results in a simplification of the data model itself and of the axioms in the semantic reference. Indeed, all data type declarations and axioms above that involve road elements and ferry connections are simply eliminated by the projection.

4.3 Semantic Transformation

In the *spatial* referencing case, a transformation from one reference system to another can involve a change of datum as well as in the coordinate system and projection. The computations necessary for an exact transformation can be very involved and are in practice often replaced by approximations with empirically derived parameters (e.g., an affine transformation where the parameters are estimated from a set of common points in both models). Fig. 2 shows a transformation between two Cartesian coordinate systems (a translation vector T and 3 rotations around the axes u, v, and w).

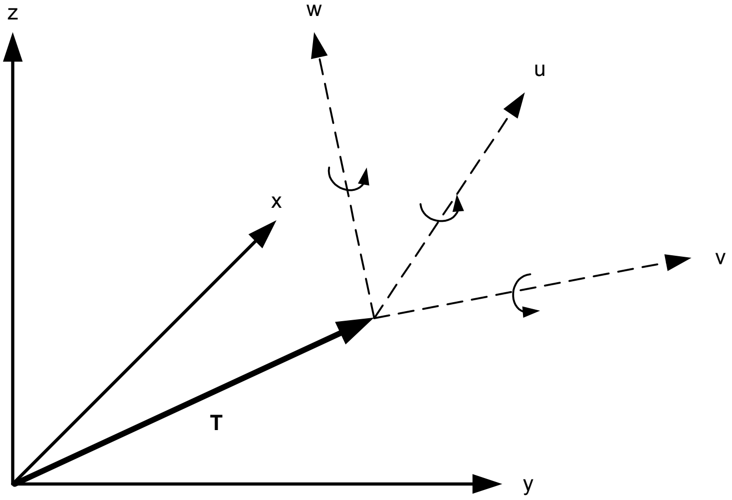

Fig. 2 Spatial transformation between two Cartesian coordinate systems.

Semantic transformations should accommodate the same generality of possible changes in datum, reference frame, and projection. Such mappings of terms from one context to another require powerful mathematical instruments of the kind found in category theory (Asperti and Longo 1991). The structures we have defined and their implementation in a higher-level functional programming language provide a relatively straightforward computational mechanism for such semantic transformations, without the need of going into theoretical underpinnings in category theory.

We present the example of exporting data from our simplified navigation model to a system that allows cars to move across edges, regardless whether these edges stand for road elements or ferry connections. Consequently, we need to introduce an axiom stating this new interpretation:

```
instance Path Edge Node Car where
  move (Car node) edge = Car (other edge node)
```

This semantic transformation leads from a system that distinguishes roads and ferries to one that does not. The latter system can recognize that the data only support route *planning*, or it can, implicitly or explicitly, ignore that limitation and use the data for route *guidance*. The interpretation of the simplified data model is satisfactory for route planning, but inadequate for route guidance: cars may be guided across rivers at ferry connections, as if there was a bridge (i.e., road element) there. Indeed, such accidents have happened (Observer 1998).

5. CONCLUSIONS

The purpose of this paper was to show the theoretical background and practical use of a close analogy between spatio-temporal and semantic reference systems. We presented an implementation of all parts of a semantic reference system for a simple navigation application. The feasibility of such an implementation in Hugs demonstrates the usefulness of both the analogy and the tool. Any equally powerful tool could be used to implement such a semantic reference system.

After this proof of concept has been achieved, many questions remain to be addressed. Among them are issues of scaling and implications for ontology design. The question whether our approach scales up to significant geospatial applications can only be answered through more extensive case studies. We are currently undertaking several such studies in the areas of emergency management and environmental planning (http://musil.uni-muenster.de).

The question of implications for ontology design is important and goes far beyond the scope of this paper. While ontologies as we know them today are a necessary ingredient of semantic reference systems, they need to be complemented by computational mechanisms that go beyond their current capabilities. Producers and users of geographic information need support for data model projections and for transformations among different semantic reference systems. Such tools are hard to build on the weak formal foundations of current ontology languages. They require concept hierarchies with sound typing, parameterized polymorphism, multiple inheritance of behavior, and higher order reasoning capabilities, as we have shown here and previously in (Kuhn and Frank 1997; Kuhn 2002). Furthermore, to relate the meaning of terms to GIS applications, entities and relationships need to be placed in the contexts of the roles they play in human activities (Kuhn 2001).

A currently debated architectural question is where and how to offer semantic information in an information infrastructure like the semantic web or some geospatial data infrastructure. The layering of ontologies coming with the idea of semantic reference systems (distinguishing upper-level from domain ontologies and semantic referencing) shows a way to address this question that has also been proposed by other authors (e.g., Fonseca et al. 2002). In the context of (web) service description languages, which is not further discussed here, this seems to favor decentralized registries over centralized approaches. Whatever the current movement toward the semantic web will bring, its applications in the geospatial communities and elsewhere will need the kind of referencing, wrapping, projection, and translation capabilities proposed in this paper.

ACKNOWLEDGMENTS

Comments from Michael Lutz and other members of MUSIL to drafts of this paper helped clarify the ideas. Support from the European Commission through the ACE-GIS and BRIDGE-IT projects, and from BMBF through the meanings project in the Geotechnologies program are gratefully acknowledged. Teleatlas provided test data in the GDF standard.

REFERENCES

A. Asperti and G. Longo (1991) Categories, Types, and Structures - An Introduction to Category Theory for the Working Computer Scientist. The MIT Press, Cambridge, MA.

F. T. Fonseca, M. J. Egenhofer, P. Agouris, and G. Camara (2002) Using Ontologies for Integrated Geographic Information Systems. Transactions in GIS 6(3).

T. Gruber (2002) What is an Ontology? http://www.ksl.stanford.edu/kst/what-is-an-ontology.html

N. Guarino (1998) Formal Ontology and Information Systems. in: N. Guarino (Ed.), Formal Ontology in Information Systems (FOIS'98), Trento, Italy, pp. 3-15.

P. Hudak (2000) The Haskell School of Expression: Learning Functional Programming through Multimedia. Cambridge University Press, New York.

ISO (2001) GDF - Geographic Data Files - Version 4.0. ISO/TC 204 N 34, Technical Report ISO/CD 2001-02-14.

ISO (2002a) ISO 19107 - Spatial Schema. ISO TC 211, Technical Report.

ISO (2002b) ISO 19108 - Temporal Schema. ISO TC 211, Technical Report.

ISO (2002c) ISO 19112 - Spatial referencing by geographic identifiers. ISO TC 211, Technical Report.

M. Johnson (1987) The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason. The University of Chicago Press, Chicago.

W. Kuhn (2001) Ontologies in Support of Activities in Geographic Space. International Journal of Geographical Information Science 15(7): 613-631.

W. Kuhn (2002) Modeling the Semantics of Geographic Categories through Conceptual Integration. in: M. Egenhofer and D. Mark (Eds.), Geographic Information Science - Second International Conference, GIScience 2002, Boulder, CO, USA, September 2002. Lecture Notes in Computer Science 2478, pp. 108-118, Springer, Berlin.

W. Kuhn (in press) Semantic Reference Systems. Guest Editorial, International Journal of Geographical Information Science.

W. Kuhn and A. Frank (1997) The Use of Functional Programming in the Specification and Testing Process. in: M. Goodchild and M. Egenhofer (Eds.), Interop'97, Santa Barbara, California, U.S.A.

A. Leick (1995) GPS - Satellite Surveying. Wiley, New York.

P. Longley, M. Goodchild, D. Maguire, and D. Rhind (2001) Geographic Information Systems and Science. Wiley, Chichester, England.

Observer (1998) BMW computer makes driver turn into drinker.

OGC (1998) The OpenGIS® Guide. The Open GIS Consortium, Technical Report.