Formalisation of Spatial Standards

Silvia Nittel

Data Mining Lab, Computer Science Dept.

University of California, Los Angeles

silvia@cs.ucla.edu

Stephan Winter

Department of Geoinformation

Technical University Vienna

winter@geoinfo.tuwien.ac.at

Extended Abstract

The problem of building and maintaining large and heterogeneous information systems — not only spatial ones — is a problem generally acknowledged by the software industry [1]. Software engineering provides different

techniques and tools for structuring the software development process into different phases. Thereby, one of the most important phases (and products) is the *specification* [2, 3]. It structures a task at hand into single actions, describes the restrictions of the actions, and captures which results are expected. However, the specification does not go into detail about *how* the actions are executed.

Specifications are the core of standardization efforts since a specification defines the underlying conceptual model of a standard (i.e. the "world" and its meaning). Standards go one step further, and specify implementation interfaces for tasks (i.e. data structures and operation signatures) to achieve interoperability between information systems. For standards, it is significant that the semantic aspects of implementation interfaces are unambiguous so that all people who build products upon the standard have the same understanding about the meaning of interfaces. This is complicated by the fact that:

- specifying experts and implementation experts often cannot communicate directly when standards are published;
- therefore, full understanding of the specification has to be derivable from the specification documentation directly;

- specifications with a status of a standard should be consistent and error-free to avoid costly changes to products.

Current specification techniques in spatial standard organizations are based on graphical object oriented modelling in UML [4, 5]; they lack formal modeling of the semantics of interfaces. The requirements in the field of spatial information systems recently lead to proposals to use algebraic specifications as an appropriate tool [6, 7, 8, 9, 10]. Algebraic specifications have mathematical clean form, and allow to capture the semantics of operations formally [11, 12, 13]. Furthermore, they are constructive, which means they are executable and their behavior can be checked when implemented using functional programming languages, and modules can be composed to more complex systems. Today, languages like Haskell are suitable tools to implement multi-sorted algebras since they are declarative, operational, and object-oriented [14]. The work of Kuhn and Frank demonstrates the usability of Haskell for capturing the semantic of expressions.

In this paper, we go one step further, and investigate the applicability, and usefulness of multi-sorted algebras and functional programming languages as a specification tool for spatial interoperability standards. To test our case, we choose the coverage section of the OpenGIS Consortiums's (OGC) spatial

interoperability standard, and modeled and implemented the specification of *OGC coverages* via Haskell. In contrast to the semi-formal UML tool chosen by OGC we were able to capture operation semantics in a formal, and less unambiguous way, and test the correctness and consistency of the specification via executable code. Certainly, a formal specification does not come without a price. The syntax of functional programming languages seems to be less intuitive to use and understand in the beginning, and is a unfamiliar tool for many people involved in the standardization process. In this paper, we discuss and evaluate the effectiveness of such a tool for the definition of spatial standards, and discuss its applicability to the real-world process of standardization. We came to the result that a functional specification would be a useful and necessary replacement and/or supplement to the verbal and semi-formal specification methods used today.

References

[1] W. W. Gibbs, "Software's chronic crisis", *Scientific American*, vol. 271, no. 3, pp. 72–81, 1994.

[2] B. Liskov and S. Zilles, "An introduction to formal specifications of data

abstractions", in *Current Trends in Programming Methodology* (R. T. Yeh, ed.), vol. 1, pp. 1–33, Englewood Cliffs, N.J.: Prentice-Hall, 1978.

[3] B. Liskov and J. Guttag, *Abstraction and Specification in Program Development*. The MIT Electrical Engineering and Computer Science Series, Cambridge, MA: MIT Press, 1986.

[4] OMG, "UML resource page", http://www.omg.org/uml/, 2000.

[5] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language Reference Manual*. Reading: Addison-Wesley, 1999.

[6] M. J. Egenhofer and A. U. Frank, "Object oriented modeling for GIS", *Journal of the Urban and Regional Information Systems URISA*, vol. 4, no. 2, pp. 3–19, 1992.

[7] A. U. Frank and W. Kuhn, "Specifying open GIS with functional languages", in *Advances in Spatial Databases* (M. J. Egenhofer and J. R. Herring, eds.), vol. 951 of *Lecture Notes in Computer Science*, (Berlin), pp. 184–195, Springer, 1995.

[8] A. U. Frank and W. Kuhn, "A specification language for interoperable GIS", in *Interoperating Geographic Information Systems* (M. F. Good-

child, M. Egenhofer, R. Fegeas, and C. Kottman, eds.), pp. 123–132, Norwell, MA: Kluwer, 1999.

[9] W. Kuhn, "Approaching the issue of information loss in geographic data transfers", *Geographical Systems*, vol. 4, no. 3, pp. 261–276, 1997.

[10] A. U. Frank, "One step up the abstraction ladder: Combining algebras — from functional pieces to a whole", in *Spatial Information Theory* (C. Freksa and D. M. Mark eds.), vol. 1661 of Lecture Notes in Computer Science, Berlin: Springer-Verlag, pp. 95-107, 1999.

[11] J. V. Guttag and J. J. Horning, "The algebraic specification of abstract data types", *Acta Informatica*, vol. 10, pp. 27–52, 1978.

[12] I. V. Horebeek and J. Lewi, *Algebraic Specifications in Software Engineering*. Berlin: Springer-Verlag, 1989.

[13] J. Loeckx, H.-D. Ehrich, and M. Wolf, *Specification of Abstract Data Types*. Chichester: Wiley-Teubner, 1996.

[14] P. Hudak, J. Peterson, and J. H. Fasel, "A gentle introduction to Haskell 98", http://www.haskell.org/tutorial/, 1999.