Formalizing Specifications for Geographic Information

Jesper Vinther Christensen
National Survey & Cadastre – Denmark and
Technical University – Denmark
jvc@kms.dk

SUMMARY

The presented work is a framework for developing production specifications for geographic information. The framework includes support for both natural language and formal language specifications, and what are important, mappings between statements in natural language and statements in formal language. The framework includes a build-in formal specification language, the High Level Constraint Language (HLCL), which is based on a first order predicate logic. HLCL can be regarded as a kind of description logic extended with some extra facilities, which makes it well suited for specifying complex rules and constraints that often occurs while specifying spatial properties. Further more HLCL is designed with a syntax that has resemblance to natural language, which makes it easier to use than most other formal specification languages. HLCL expressions can be translated into SQL, and used to validate if produced information conforms to stated requirements. In this paper we focus on HLCL's capabilities for domain and conceptual modelling, while we refer to (Christensen & Johnsen 2005) for details on using HLCL for validating produced information, and (Fischer-Nilsson & Johnsen , forthcoming) for the mathematical foundation of HLCL.

KEYWORDS: *Formal specification, ontology, domain modelling, data content specification.*

INTRODUCTION

Specifications are created to describe a vision developed and owned by a group of individuals. Without formulating knowledge, ideas, and decisions in a specification, an in depth understanding of the problem domain will never be achieved, and a possible design will be based on un-solid ground. In this way the process of developing specifications form consensus among the individuals, groups, or organizations that have interest in the geographic information produced from the specification. A second point that motivates the development of specifications is to be able to communicate, in detail, what content the information produced from the specification can be expected to have. In this way specifications can be regarded as detailed meta data, or more general meta data may be derived from a specification. The content of specification is influenced by a number of factors. Ideally users requirements are decisive when the content and structure of a geographic data set is designed, but naturally other factors influence a specification, amongst others politics, cultures, and traditions, the nature of domain, and available technologies and resources.

The difficulties in specifying geographic information arise from two sources, one: Geographic information is about "real things" like lakes and forests. Every one who had tried to define what a forest is, or to determine the boundary of a lake, or to answer questions like "can a lake be a part of a forest?" knows how hard this is. The second source contributing to the complexity of specifying geographic information is large the amount of information needed to describe a data collection, like a topographic map. Specifications may be covering 300-500 pages, including detailed descriptions for maybe 50 object types or more.

If the information produced from a specification should be consistent and homogeneous it is imperative that the specification is clear and unambiguous, and leaves as little room for interpretation as possible. A systematic approach for identifying and denoting phenomena in the real world and their representations, and a predefined structure for organizing and classifying specification elements would help achieving this goal. This is what this paper is about: A framework for developing

production specification for geographic information is introduced. The framework has four important properties:

- Distinction between domain concepts and the representation of these.
- Supporting statements in both natural languages and formal languages
- Including a build-in formal specification language, called HLCL.
- Structuring the detailed descriptions needed to produce information from the specification

The need for specialized languages and frameworks dedicated the specification of geographic information has been acknowledge by several authors and projects: the COSIT laboratory at IGN has proposed a model for structuring data contents specification (Gesbert 2004 & Mustiére, 2003). The MADS data model provides a number of icons, with which classes can be labelled to symbolize spatial properties and relations (Parent 1998). In (Friss-Chrisensen and Christensen 2004) an extension of UML is suggested to requirements relations to acceptable quality levels and with which quality parameters the quality must be described.
The object constraint language, OCL (Object Management Group 2003) has been tested for its capabilities for writing constraints for geographic information (Casanova 2000).

SPECIFYING GEOGRAPHIC INFORMATION

The approach taken in this paper for structuring specifications for geographic information is to regard a specification as a set of statements. Each statement can be classified according to the role it plays in the specification. In the following sections three angles of the classification of statements are discussed. First a representation system is introduced grouping statements in descriptions of a domain model and descriptions of a representation of a domain as a conceptual model. Second statements that bind the domain model together with the conceptual model are introduced, and third statements are classified to be either informal or formal. Finally a grammar for writing and classifying statements is introduced.

Representation model

A specification constitute what is called a nominal ground or universe of discourse, and can be regarded as a mechanism that points out entities in a domain and defines how these are represented as objects. In general a specification is composed of statements on the form: *"if something holds in the domain, then something must hold in our representation of the domain"*. Therefore, to be able to develop clear and unambiguous specifications, a systematic approach for defining and handling concepts for both "real world entities", their representations, and statements binding concepts and representations together is needed.

We will use the term *domain* with the meaning given by Jackson, Shlaer and Mellor (Jackson 1995 & Shlaer 1992): a part of the real world that is interesting for a particular problem. Domains that are interesting for the specification of geographic information include geographic entities. A geographic entity is a real world phenomenon with spatial properties, which in a given context can be distinguished from all other geographic entities. An entity is characterized by a number of properties. Each property may have a value determined by an observation. Distinguishing between what are entities and are properties can in practice be challenging. In general an entity is a phenomenon that should be exposed as an individual, while properties are unary predicates and functions, e.g. the house is used for living, and the height of the house is 10 meters, describing a particular entity.

While domains deal with individuals, domain models concern sets of individuals. Geographic entities can be ordered in sets according to properties, e.g. material, function, usage, and spatial relation to other entities, but also location, orientation, size, and shape play a role when classifying geographic entities. A domain model is a conceptualisation of a domain, including taxonomic and ontological descriptions. Each concept or set of individuals is denoted with a term giving it a name.

Apart from concepts with names, a domain can be described by a set of assertions. Assertions are used to describe the domain in detail. Assertions can for example describe relations among concepts, like part-whole relations and associations, but also complex relations can be described.

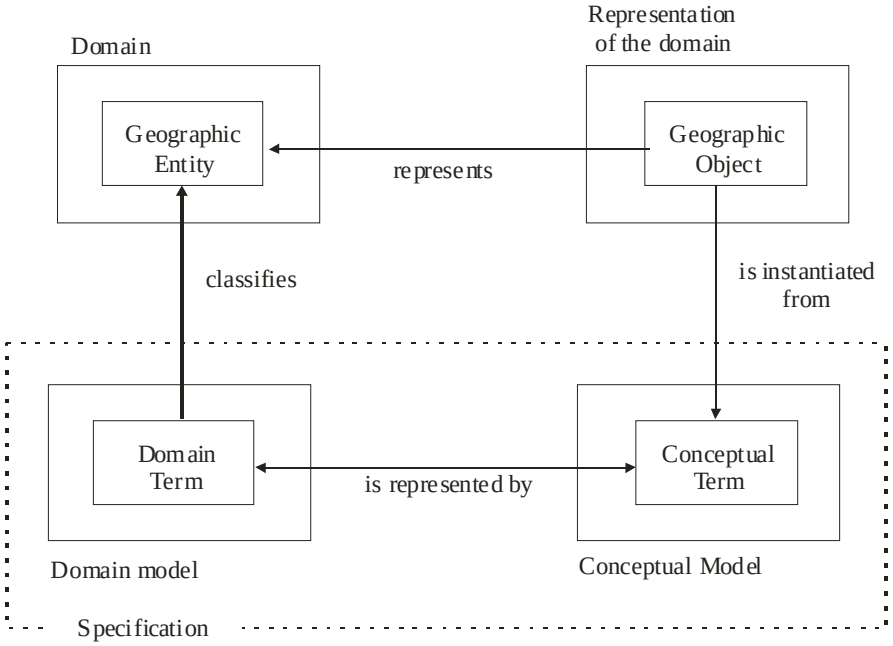

Figure 1: Representation model

A conceptual model expresses design decisions. Based on acquired requirements and the domain knowledge, the conceptual model formulates the structure in which entities must be represented, and the constraints that the represented entities must comply to. Conceptual models contains information about the object types in the form of attributes and relationships between the object types, there is no information about the meaning of the object types, but only a specification of what kind of information will be given about the entities represented by the object types in question.
Terms describing included in the domain model and terms included in the conceptual model can be related to define how entities in the real world is represented as objects in a collection. The relations between domain terms and conceptual terms also describe how generalization of geometric representation is made and how the different data sources are interpreted in order to extract the wanted information. For example a buildings outline is drawn from a digital image or a classification of vegetation is made.

Informal and formal descriptions

What is characterizing a design process is that the knowledge the designers acquire about the problem domain increase over time. Specifications evolve over time, starting as weak ideas and problem descriptions they grow into stable requirements, and finally turn into detailed designs that can be used as the fundament for an implementation. The first version of a specification may be a rough sketch, written in plain English, which in following design moves is enhanced and expanded, resulting in a precise and formal specification. A specification language should include structures to include both

informal and formal specification elements, and be able to help the designers to keep track of the relations among the various specification elements.

Grammar – Basic concepts

The fundamentalspecification elements in a specification developed in GeoSML are **Statements**. Statements can be written in several languages, both natural languages and the built-in formal specification language (High Level Constraint Language, HLCL). **Assertions** are statements describing the domain, while **Constraints** are statements describing the conceptualization of the domain. **Terms** that are important in the descriptions can be marked and defined, and it is possible to form **Entity Type** with

Properties and **Object Types** with **Attributes**, describing the domain and the conceptualisation of the domain respectably. Statements can be labelled with a **Statement type** describing the purpose of the statement. For both model types it is possible to describe relationships, in domain models between entity types, and in the conceptual models between object types. Assertions and Constraints can be linked together. This is done if a constraint implements rules the reflex the domain knowledge captured by the assertion. Entity types and Object types can also be linked using the **is represented as** relation. The following grammar defines the basic elements of GeoSML in a context free grammar in EBNF.

⟨statement⟩::=(⟨language⟩(⟨exp⟩)*)*(⟨formalexp⟩)*

⟨exp⟩::=string

⟨formalexp⟩::=⟨domain relation⟩ |⟨conceptual relation⟩ |⟨constraint⟩ |⟨repexp⟩

⟨entity type⟩::=**"Entity type"**⟨entity typeID⟩
"Name:"⟨domain termID⟩
"Properties:"(⟨propertyID⟩⟨domain termID⟩)*

⟨assertion⟩::=**"Assertion"**⟨assertionID⟩⟨statement⟩

⟨domain relation⟩::=⟨entity typeID⟩⟨domain relspec⟩⟨entity typeID⟩

⟨domain relspec⟩::=**"is-a"** |**"part-of"** |⟨named domain relation⟩

⟨named domain relation⟩::=⟨domain termID⟩

⟨domain term⟩::=⟨domain termID⟩(⟨language⟩⟨term⟩)*

⟨object type⟩ ::= **"Object Type"** ⟨object typeID⟩
"name:"⟨conceptual termID⟩
"Attributes:" (⟨attributID⟩⟨conceptual termID⟩⟨datatype⟩)*

⟨constraint⟩::=**"Constraint"**⟨constraintID⟩⟨statement⟩

⟨conceptual relation⟩::=⟨object typeID⟩⟨conseptual relspec⟩⟨object typeID⟩

⟨conceptual relspec⟩::=**"is-a"** |**"part-of"** |⟨named conceptual relation⟩

⟨named conceptual relation⟩::=⟨conceptual termID⟩

⟨conceptual term⟩::=⟨conceptual termID⟩(⟨language⟩⟨name⟩)*

⟨assertion mapto constraint⟩::=⟨assertion⟩ **"is implemented by"**(⟨constraint⟩)*

⟨repexp⟩::=⟨entity typeID⟩ **"is represented as"** ⟨object typeID⟩
"setting" (⟨attributID⟩ **"to"** ⟨value⟩)*

Figure 2: Grammar for the basic specification elements

FORMALIZATION OF STATEMENTS USING HLCL

The High Level Constraint Language (HLCL) (Christensen & Johnsen 2005) was developed to bridge the gap between constraints or business rules formulated in natural language and their implementation using e.g. SQL. In this paper HLCL is embedded in framework defined by the above grammar and few minor justifications of the original grammar, improving HLCL's capabilities for conceptual model.

HLCL overview

HCLC-expressions use two basic constructions: the "all-must" and "no-may" expressions.

all residential area must contain solely residential buildings

The above constraint makes use of the "all-must" construction. The **"solely"** keyword expresses that residential area should only contain residential buldings and nothing else. Similarly to the "**solely**" keyword, HLCL has an "**all**" keyword, which expresses that the relational path should be fulfilled for all the classes, e.g. "contain all building" expresses that all buildings should be contained. Finally there is the option using a numerical quantifier "**at least/at most/exactly n**", e.g. "contain at least 5 building" expresses that at least five buildings should be contained.

no lake may contain building

The above constraint expresses that "*no lake may contain any building*". In the above example the inverse top-construction "**no-may**" is used to express class disjointness. The "**contain building**" fragment specifies a relational path in the conceptual model. Relational paths are expressed by a series of relations and classes; they are implicit existentially quantified in HLCL, such that the path above is understood as "*contain at least one building*". Paths can be of any length, one simply adds more relations and classes, and multiple paths can be bundled together by the "**or**" disjunction and the "**and**" conjunction operator.

HLCL allows the usage of user-defined functions, various arithmetical relationships, and variables. The example in one of the following sections will illustrate how these constructions can be used in a specification.

HLCL grammar

The complete grammar for the concrete syntax of High Level Constraint Language is included in figure 3. More examples of HLCL expressions are given in next section.

⟨constraint⟩::=**"all"** ⟨class exp⟩ **"must"** ⟨class exp⟩ |
"no" ⟨class exp⟩ **"may"** ⟨class exp⟩

⟨class exp⟩::=⟨conceptual entityID⟩ [⟨rel class⟩] |⟨function⟩ |⟨varclass exp⟩

⟨varclass exp⟩::=⟨conceptual entityID⟩ ⟨variable⟩ [⟨rel class⟩]

⟨rel class⟩::=**"("** ⟨rel class⟩**")"** |
⟨relation⟩ [⟨int quant⟩] ⟨class exp⟩ [⟨operator⟩ ⟨rel class⟩] |
⟨relation⟩ ⟨varclass exp⟩ [⟨operator⟩ ⟨rel class⟩] |
⟨attribute⟩ ⟨value⟩ |
⟨attributeID⟩ ⟨numerical relation⟩ ⟨integer⟩ ⟨value⟩

⟨int quant⟩::=**"all"** |**"solely"** | ⟨numerical relation⟩ ⟨integer⟩

⟨operator⟩::= **"and"** | **"or"** | **"andnot"** | **"ornot"**

⟨numerical relation⟩::=**"exactly"** |**"at least"** |**"at most"**

⟨variable⟩::= **"A"** | ... |**"Z"**

Figure 3: Grammar for the Híigh Level Constraint Language

USING THE FRAMEWORK

To illustrate how the specification framework is used this section includes a simple example of a specification. The example concerns the specification of the building object type in a topographic map. As a visual guidance the domain model and conceptual model is included as ER-diagram in figure 4 and 5.

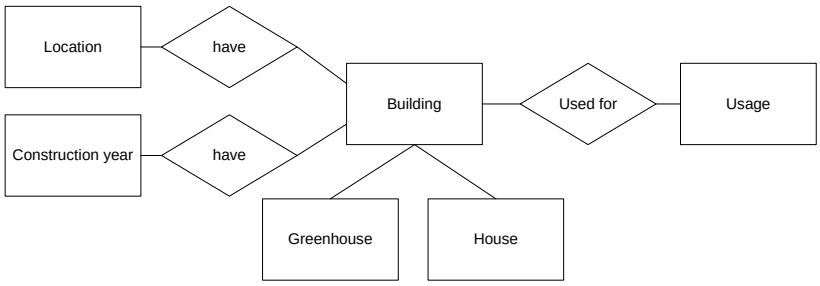

Figure 4: Domain model represented as an ER-diagram

The domain model has five domain terms: Building, Greenhouse, House, Location, Construction year, and Usage. Greenhouse and house are specialization of Building. Location and Construction year are attributes, and finally is there an association between Building and Usage named Used for.

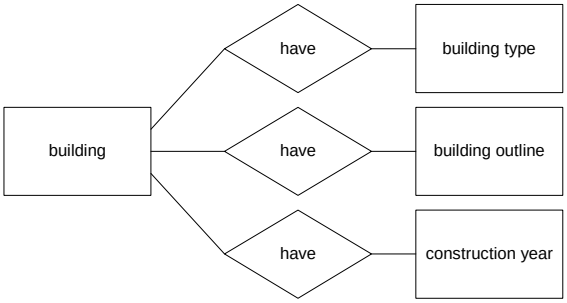

Figure 5: Conceptual model represented as an ER-diagram

The conceptual model includes four conceptual terms: building, building type, building outline, and construction year. The last three mention terms are all attributes of building.

The following specification s

Domain model

Domain term in English: **Building**
in Danish: **Bygning**
Domain term in English: **Location**
Domain term in English: **Construction year**
Domain term in English: **Greenhouse**
Domain term in English: **House**
Domain term in English: **Usage**
Domain relation: **Greenhouse is-a Building**
Domain is in HLCL: **House is-a Building**
Entity Type **Building**
Name: **Building**
Properties:
Location
Construction Year
Domain Relation: **Building must be used for Usage**

Conceptual model

Conceptual term in English: **building**
Conceptual term in English: **construction year**
Conceptual term in English: **type**
Conceptual term in English: **building outline**
Object type
Name: building
Attributes:
construction year: date
type: string
building outline: geometry
Constraint
in HLCL: **All building must have building outline**
Constraint
in HLCL: **All building must have atleast 0 construction year and have atmost 1 construction year**

Constraint in HLCL: **all building of_type house must have construction year**
Constraint in HLCL: **no building may contain building**
Constraint in HLCL: **no building may overlap building**
Constraint concerning building in English: **If two buildings of the same type is neighbor then the z-difference between the two building must be larger than 5 meters**
in HLCL: **all Building A hastype T neighbor Building B hastype T must havezdifferencebiggerthan(A,B,5)**

Mapping domain model and conceptual model

Representation Rule concerning Building in English: **Greenhouse is represented as building setting type to "greenhouse", building outline to Location, construction year to construction year**
Representation Rule concerning Building in English: **House is represented as building setting type to "house", building outline to Location, construction year to construction year**
Representation Rule concerning Building in English: **Two neighbor Buildings having a height difference smaller that 5 meter must be represented as one building**
Representation Rule concerning Building in English: **building outline is registered on the roof overhang/eaves of the building**
Representation Rule concerning Building in English: **Building outline must be registered as closed polygons with a common start and end point.**
Representation Rule concerning Building in English: **building outline must be registered on the outer extremity of the foundation or ruin.**
Selection Rule in English concerning Building in English: **As a general rule, all building larger than 25 sqm must be registered in their fundamental form.**
Representation Rule concerning Building in English: **buildings outline must be registered with as few points as possible but in such a way that the difference between the actual sequence and the registered sequence is never larger than 1 m in plan and elevation.**
Representation Rule concerning Building in English: **All Building corners must be registered. However, Buildings with overhangs and extensions with a side length of less than 3 m and an area smaller than 10 sqm must not be registered.**
Selection Rule concerning Building in English: **Agricultural buildings smaller than 100 sqm in connection with farms, that are considered to be used for habitation, must not be registered.**
Representation Rule in English concerning Building in English: **Houses built together must be registered as one building.**

CONCLUSION AND FUTURE WORK

The framework presented in this paper shows how production specifications for geographic information can be organized according to a predefined structure. By introducing the both domain models and conceptual model we enable the specification designers to distinguish between that is knowledge about the problem domain and what are design decisions forming the wanted representation of the domain. It is our hope that designers will be more aware of how to write definitions and designations using the suggested framework, and that the classification of statements can be used as guidance when writing detailed description explaining how entities must be represented in the data collection.

Currently a work on developing a case-tool that helps the users to develop domain models and conceptual models, and to formulate HLCL statement, is under development. The intention is that this case-tool will make HLCL even more accessible to non-programmers, and enable designers to handle the large information needed to specify for example topographic maps, by supplying access to the specification to the various specification elements though predefined views. This work also includes a XML based definition of the framework.

BIBLIOGRAPHY

Baader, Franz ed., 2003 The Description Logic Handbook: Theory, Implementation and Applications. Cambridge

Bassiliades N,.Gray P (1995) Colan: A functional constraint language and its implementation. Data Knowledge Eng. 14, 203-249.

Casanova M, Wallet T, D'Hondt M (2000) Ensuring Quality of Geographic Data with UML and OCL, in proceedings of the 3rd International Conference on The Unified Modeling Language, Vol. 1939 of Lecture notes in computer science, Springer, pp 225 - 239

Christensen, Jesper V. and Johnsen, Mads,. 2005 Formal Constraints for Geographic Information in Proceeding of ISD 2005: Information System Development. Springer.

Fischer-Nilsson Jørgen , Johnsen Mads. (Forthcoming) A high level logical-algebraic constraint checking language compiling into database queries.

Garshol, Lars M., BNF and EBNF: What are they and how do they work? http://www.garshol.priv.no/download/text/bnf.html

Hoel E, Menon S, Morehouse S (2003) Building a Robust Relational Implementation of Topology. International Symposium on Spatial and Temporal Databases 8.

Horrocks Ian et al. 2003 SWRL: A Semantic Web Rule Language: Combining OWL and RuleML. URL: http://www.daml.org/2003/11/rules-all.html

Johnsen M (2005) A High Level Interface to Databases, with Application to GIS (2005), Danish Technical University of Denmark.

Mustiére S, Gesbert N, Sheeren D. (2003) A Formal Model for the Specification of Geographic Databases. International Workshop on Sematic Processing of Spatial Data.

National Survey & Cadastre - Denmark (2003) TOP10DK data content specification version 3.2 (in Danish).

Object Management Group (2004). UML 2.0 OCL Specification

Parent, C. et al. 1998 Modeling Spatial Data in the MADS Conceptual Model Proceedings of the 8th International Symposium on Spatial Data Handling