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

MODELING PROCESSES DEFINED BY LAWS

Gerhard Navratil, Andrew U. Frank

Institute for Geoinformation, Technical University Vienna,
Gusshausstr. 27-29, A-1040 Vienna

1. INTRODUCTION

Laws regulate our everyday life. It is rather difficult to think of actions that are not influenced by laws. Driving a vehicles, buying a piece of bread, or smoking are only some examples where laws influence the way we act. Laws also influence geographic information systems. Definition of classification schemes, procedures and critical values are often written down in laws. We have to make sure that evaluations done by geographic information systems do not only reflect the real world situation but also the legal parts. A method to do this is formalization of the law texts.

Moreover laws should be consistent. Laws should have no contradictions, i.e. they must not do both, allow and prohibit something. This is not only valid within a specific law but also between different laws in one country or between different countries. Within the European Union, for example, several definitions must be uniform to grant homogeneous and fair financial support. Formalization of the laws allows checks on the consistency.

We formalized the Austrian land registration law (GBG55). This law has some advantages: It is a rather old law and underwent only minor changes since its creation in 1871. Therefore, problems with the law should have been detected and eliminated by now. The land registration law is a strict law which specifies a small, encapsulated system. These advantages allowed the creation of an algebraic model, which can be found in [1].

2. LAWS

An important question is how to compare legal situations in different countries. Work on this question include logic based approaches [2-4], ontologies [5-7], and representation languages [8].

Pure logic has a nice property. It shows clearly the connection between reason and effect. It also shows the same structure as the law text itself and therefore it is easy to create a link to the original text. However, pure logic is difficult to read for persons without training in logic expressions.

Representation languages and ontologies have problems when dealing with comparison. Representation languages like METALex represent the legal text in a clear structure. They even provide the possibility of different languages for the text. However, the problem is that they do not show how the law works and how different rules fit together. Ontologies concentrate on these things. They are designed to illustrate the meaning of legal definitions. Their problem is the connection to the law text. It is difficult to see where the rules can be found. It would be necessary to have a solution that combines the advantages of the three approaches:

- connection to the original law text,
- usage of different languages,
- representation of connections between paragraphs.

A method which combines all these advantages is the method of algebraic specifications. This method has been used for different tasks like modeling wayfinding tasks [9, 10], software specification [11], or GI-Systems [12,13].

3. ALGEBRAIC SPECIFICATIONS

Advanced specification techniques emerged from the need of better development tools. The process of designing correct software is a quite difficult task and bugs often result in financial losses [14]. Therefore, different methods for better specification methods have been produced. One of these methods is algebraic specification, which fulfills the following conditions [15]:

- leads to adequate programs, i.e. programs that solve the customers problem,
- leads to correct programs, i.e. programs that are free of bugs and thus behave the way the programmer wants them to behave,
- leads to programs whose maintenance is easy, i.e. programs that can be easily corrected or modified without introducing errors.

Modeling complex systems with algebras has some advantages. Algebras describe abstract classes and their behavior. Abstract specifications allow reusing the code which allows the designers to concentrate on the specific problems of the program without having to deal with standard tasks for too long. Using different classes allows the separation of the problem into smaller parts which can be modeled independent from each other. A combination of these algebras as described in [16] then leads to the specification of the whole problem. Algebraic specifications are also language independent, although they require a formal language which is capable of expressing algebraic style specifications. The purpose of a formal specification is to formally describe the behavior of objects. Algebraic specifications were introduced to describe data abstractions in software design [11].

4. SIMILARITIES BETWEEN LAWS AND ALGEBRAS

Both laws and algebras describe ideas. Laws and algebras are methods for the definition of models following specific ideas. Although the language used in the two methods is different both methods use comparable tools, i.e. subdivision, abstraction and axiomatization.

4.1 Description of a Model

The definition of a model requires a language for the description of the elements of the model and the way they interact. Laws use natural language for this purpose whereas algebras rely upon mathematical expressions. In spite of the different languages both methods result in a model.

4.2 Subdivision into Small Parts

Both laws and algebraic specifications split the problem into small parts and describe these parts. A legal system uses different laws to regulate different legal aspects. The same is done with algebras. Different aspects are treated by different algebras even if the object is the same. A river, for example, can be seen as a waterway, a barrier (for a street), and a type of waters. Each of these aspects allows different actions and has different parameters.

There is further subdivision within a law. Laws must define terms and specify methods, processes, and effects. This requires a subdivision of the law in smaller blocks which are then organized by paragraphs and enumerations.

4.3 Abstract Description of Rules

Laws as well as algebraic specifications describe situations as abstract as possible. Paragraphs in laws consist of legal situation and legal consequence. A legal situation is a typical behavior or situation, the legal consequence is the resulting effect. The effect may be a claim or a punishment. Sometimes, like in § 1 GBG55 there are no legal consequences because the paragraph only defines a term. The definition of the legal situation can be done by prototypical examples or by specifying typical qualities of the situation. The right of ownership, an affected piece of land, two or more involved persons, and the action of registration, for example, specify the situation where somebody bought a parcel and wants to register his ownership.

4.4 Axiomatic Definition and Description

Both laws and algebras define operations by axioms. Laws use specific terms and define the connection between these terms by general statements which can be seen as axioms.

4.5 Links

Laws contain links to other laws. These links are written explicitly in the law text and are helpful for the reader because they show connections to other laws. § 9 GBG55 is an example for such a link. It defines which types of rights can be registered in the land register. It states the general group of private rights and three public rights. The definition of these public rights is not written in the land registration law because they are defined in the Austrian civil law code. Therefore, the land registration law only contains a link to the appropriate paragraphs in the civil law code.

Sometimes, links are not written explicitly in the law text. In commented prints of laws you often find comments like ‘see also §…’, ‘for more information see § …’, or similar. These comments show links to other laws which could be missed by the reader. These links could be links to laws that have been used to clarify situations or links to laws created after the law at hand.

Algebras require no explicit links. Algebras use the definitions of other algebras by naming the algebras used. These definitions will then be used within the algebra. It is not necessary to mention the source because mathematical definitions are always ‘known’ within the context of mathematics.

5. DIFFERENCES

There are also differences between laws and algebras. Laws define rules for human coexistence and try to motivate the humans to obey these rules. So the rules are used in the real world and the use of the rules is not part of the law. Algebraic models, however define a mathematical system, which is only used within the context of mathematics.

5.1 Imprecise Definitions

Laws often use imprecise definitions. The two reasons for this are:

- avoidance of explicit values: Laws create definitions which are valid for a long time. Statements on values of objects collide with the fact that these values change over time. This is not only valid for the value itself but also for the relation to other values. An example is the value of land. In the early 19th century the value of agricultural land was higher than the value of residential land. Nowadays the contrary is true. Imprecise definitions like ‘minor land registration aspects’ provide a solution for this problem.
- latitude for decisions: Laws cannot provide definitions for all possible cases. It is often necessary to extract the principle ideas of a law and act accordingly. However, this is only possible if the law leaves room for variations of the decision. A simple example is the height of punishment. Rather often laws define a minimum and a maximum boundary (e.g. 5 to 10 years of prison). This provides the judge with the possibility to adjust the punishment to the situation and to take facts into consideration which are not part of the law.

Algebras provide unambiguous rules for the operations. Therefore, the results of the operations are precisely defined. Adding two integer numbers, for example, should leave no room for variations because different mathematicians should come to the same result when using the same numbers. Latitude for decisions is not necessary because the rules are applied to precisely defined situations. It is not possible, that rules for integer numbers are applied to real numbers, for example.

5.2 Organizational Contents

Laws as a part of the organization of a country are subject to changes. A smooth transition from one set of rules to another set of rules requires rules itself. Such transition rules define for example the date from which the new rules are in effect or the authority responsible for implementing the rules. Algebras do not need such organizational rules. An algebra either exists of it does not exist. An algebra does not have a start date or end date for it's validity. Besides, an algebra is handled by mathematical rules and requires no special organization for execution.

Laws have a temporal aspect. Law changes because new laws or changes in laws also change law as a whole. The changes take place as an abrupt transition at a specified point in time and not continuously. The point in time is the date from which the law is valid and this date is named in the law. Laws often replace older versions of the same law or parts of other laws. These older laws become invalid, a fact which must be mentioned in the law.

Laws require organizations responsible for enforcing the law. Laws require transfer into reality. Laws are the rules for the coexistence of humans. They are worthless if they only exist on paper and are ignored in reality. Thus laws require organizations to implement the rules. The Austrian ministry of justice, for example, is responsible for the Austrian land registration law.

Algebras do not need such a transfer. Algebraic models shall reflect reality but it is not necessary to implement them. Therefore there is no need for organizations and consequently there are no rules for implementation.

5.3 Contradictions

Laws may contradict each other without attracting immediate attention. It may happen that for a specific situation two laws ask for different actions. This would be a contradiction which will only be visible if this specific situation occurs and the situation is discussed by lawyers. The contradiction is irrelevant, however, if the situation cannot occur. The Austrian civil law code contains rules about servitude. Since servitude cannot occur nowadays contradictions to other rules do not matter.

Algebras do not allow contradictions. An algebra consists of unambiguous mathematical definitions. Contradictions violate the unambiguousness. It is a mathematical contradiction if there are two different results for one situation because a mathematical operation always has a unique result if the range for input and output is set properly. Therefore it is not possible to model contradicting legal rules. Contradictions in laws must be solved prior to formalization.

6. CONCLUSIONS

Laws and algebras have some important aspects in common. Both describe models and do so in an abstract way. The main differences are imprecise definitions, organizational aspects, contradictions, and, of course, the language. The language of algebras is the mathematical language and provides a platform of conversation without national limitations. Finding contradictions during the formalization process improves the quality of the law and even the knowledge that there are no contradictions is important. Organizational aspects are not necessary in algebraic models and can be ignored. The only problem which needs work comes from imprecise definitions.

The paper presented a comparison between laws and algebras. Some of the benefits of algebraic models have not been addressed. For example algebraic models are executable (if specified in a suitable programming language) and if software is necessary to put the law into action, the algebraic model provides the core of the software.

Of course there are open questions. All of the examples are taken from a strict law – the Austrian land registration law. More general laws like traffic law might cause more problems. However, these questions should be discussed jointly by lawyers and mathematicians.

7. BIBLIOGRAPHICAL REFERENCES

[1] Navratil, G., *Formalisierung von Gesetzen*. GeoInfo Series, ed. A.U. Frank. Vol. 25, Vienna: Institute for Geoinformation, Vienna University for Technology, 2002.

[2] Lachmayer, F., *Grundzüge einer Normentheorie*, Berlin: Duncker & Humblot, 1977.

[3] Sergot, M.J., F. Sadri, and e. al., «The British Nationality Act as a Logic Program», *Communications of the ACM*, **29**(5), 370-386, 1986.

[4] Morscher, E., *Was sind, war tun und was können Formalisierungen im Recht.*, Salzburg: Forschungsinstitut Philosophie/Technik/Wirtschaft, 1988.

[5] Valente, A., *Legal Knowledge Engineering; A Modelling Approach*, Amsterdam, Netherlands: IOS Press, 1995.

[6] van Kralingen, R.W., *Frame-Based conceptual Models of Statute Law*, The Hague, Netherlands: Kluwer International, 1995.

[7] Visser, P.R.S., *Knowledge Specification for Multiple Legal Tasks; A Case Study for the Interaction Problem in the Legal Domain*, The Hague, Netherlands: Kluwer International, Law, 1995.

[8] Boer, A., R. Hoekstra, and R. Winkels. «Proposal for a Dutch Legal XML Standard», *Proceedings of the EGOV 2002*: Springer Verlag, Berlin, 2002.

[9] Raubal, M., *Agent-based Simulation of Human Wayfinding: A Perceptual Model for Unfamiliar Buildings*, GeoInfo Series, ed. A.U. Frank. Vol. 26, Vienna: Institute for Geoinformation, Vienna University of Technology, 2001.

[10] Hochmair, H., *The Wayfinding Metaphor - Comparing the Semantics of Wayfinding in the Physical World and the WWW*, Institute for Geoinformation, Vienna University of Technology, 2002.

[11] Guttag, J.V., E. Horowitz, and D.R. Musser, «The Design of Data Type Specifications», *Current Trends in Programming Methodology*, 60-79, 1978.

[12] Frank, A.U. and W. Kuhn, «Specifying Open GIS with Functional Languages», *Advances in Spatial Databases (4th Int. Symposium on Large Spatial Databases, SSD'95, in Portland, USA)*, 184-195, 1995.

[13] Winter, S. and S. Nittel, «Formal Information Modelling for Standardization in the Spatial Domain», OpenGIS Consortium, to appear.

[14] Gibbs, W.W., «Software's Chronic Crisis», *Scientific American* **271**(3 (September 1994)), 72-81, 1994.

[15] Loeckx, J., H.-D. Ehrich, and M. Wolf, *Specification of Abstract Data Types*, Chichester, UK and Stuttgart: John Wiley and B.G. Teubner, 1996.

[16] Frank, A.U., «One step up the abstraction ladder: Combining algebras - from functional pieces to a whole», S*patial Information Theory - Cognitive and Computational Foundations of Geographic Information Science (Int. Conference COSIT'99, Stade, Germany)*, 95-107, 1999.