Computer Aided Conformance Checking

Marcel de Waard[1]

Abstract

In the building process a number of problems exist with regard to building regulations, causing the conformance checking process to be an island in the building process. This paper discusses an approach that enables us to perform computer aided conformance checking and to integrate the conformance checking process in the building process. The approach is based on the use of product models. This paper discusses theoretical as well as implementation aspects. The paper is an extended abstract of the Ph.D. thesis written by the author [De Waard 92].

Introduction

The conformance checking process of building designs against building regulations is an activity that has to be performed during the building process. It is the point in the building process where a building permit is to be obtained, so that the actual construction of the building can begin.

The conformance checking process is still done by hand by the building authorities. If the building authorities use computers at all, the computers are not integrated with CAD systems or the like, making the information transfer from the design team to the building authorities and vice versa still done by means of paper documents. Reasons for not being able to integrate CAD systems with regulations software, if they exist, are given by Eastman [Eastman 89]. According to Eastman the functionality of CAD systems is limited, due to what he calls a lack of 'architectural semantics'. Meaning that CAD systems should be able to define and manipulate architectural objects like rooms, walls, doors, etc. instead of lines, points, etc.

To be able to support the building authorities in their work, an approach is described with which computer aided conformance checking can be achieved. The approach proposes the use of product models [De Waard 90]. The approach requires [De Waard 92]:

- an information model of residential buildings,
- an information model of building regulations, and
- a link between both information models.

The information model of residential buildings contains architectural semantics Eastman speaks of. This information model can also be used as a neutral exchange medium for information exchange between the design team and the building authorities and vice versa. The model can be seen as a view independent information model. The information model for building regulations considers building regulations as consisting of two things. A model of the objects that are subject of the regulations and constraints which limit the attributes and characteristics of, and relationships that can hold between, these objects. The building regulations considered are those stated in the new national building code called the

[1]Delft University of Technology, Faculty of Informatics, Department of Information Systems,.P.O. Box 356, 2600 AJ Delft, The Netherlands; TNO Building and Construction Research, Department of Computer Integrated Construction, P.O. Box 49, 2600 AA Delft, The Netherlands.

Building Decree, where only the requirements for residential buildings to be build are considered. The link between the information model of a building and that of the building regulations describes the mapping of the first information model on the second. The model of the objects that are subject of the regulations form a building regulations view on the building model. By describing how this view can be derived from the view independent model, computer aided conformance checking can be achieved.

The paper will discuss each of the three constituent parts in some more detail, both from a theoretical, conceptual and from an implementation point of view.

Information model for residential buildings

There are two ways of looking at a building, also seen by CIB-W74. The one is to look at a building as a space system consisting of spaces, the other is to look at a building as a building elements system consisting of building elements. The space system consists of a hierarchy of mutual connected spaces. The spaces considered depend on the building that is modeled. However, the space system can not be considered without also considering at least part of the building elements system, because part of the building elements system, like walls, floors and ceilings, make up the spaces from the space system. This is why an integration of the two systems is to be established. This is done by introducing an artificial entity type called 'space boundary'. A space is now always bounded by a space boundary and a space boundary can be formed by a separation structure. The space boundary can be used to relate information to a separation structure that is only important to one side of that separation structure. Information we can think of is wall finishes. The space boundary is also used to relate supply connection points, the end points of supply systems, to a space and a separation structure. Openings are related to separation structures instead of to the space boundary because openings are important to both sides of a separation structure. The kernel of the information model for residential buildings looks as depicted in the NIAM [Nijssen 89] model of figure 1. The kernel can be further specialized, by specializing the entity types from the kernel and as such gradually introduce more and more architectural semantics into the model. The kernel as depicted in figure 1 can also be used for other buildings, as can be seen in Van Nederveen [Van Nederveen 91].

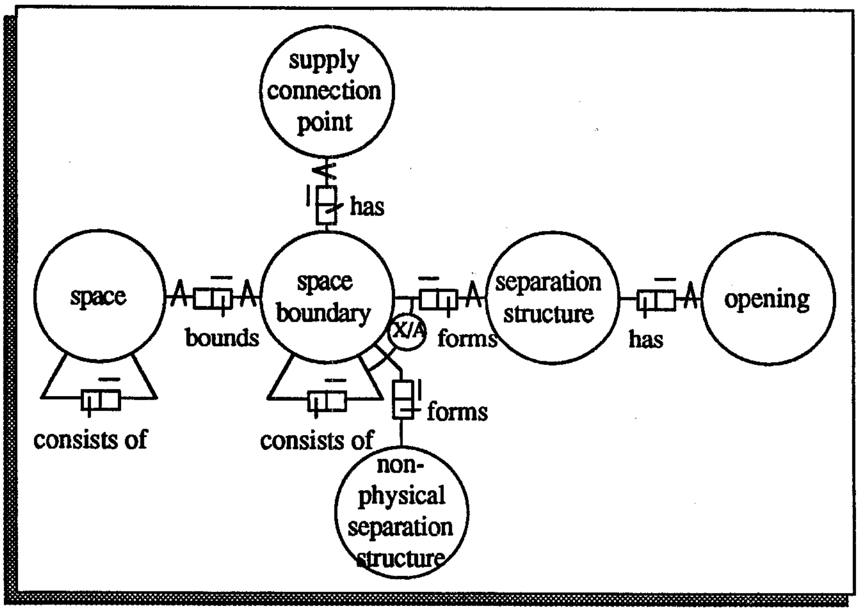

Figure 1 Kernel of the product type model for residential buildings

On this higher semantical level constructs from the GARM [Gielingh 88] are used. The GARM proposes to distinguish each product (called Product Definition Unit or PDU) in at least two stages: the 'As Required' stage, called FU, from Functional Unit and the 'As Designed' stage, called Technical Solution (TS). The FUs collect all the requirements as they come from different sources (client, regulations, company resources, etc). The GARM also proposes a hierarchical decomposition that closely follows the actual design process, namely a so-called TS-FU decomposition. Each FU can be fulfilled by one or more alternative TSs. Sometimes a TS can be bought, sometimes not. If a TS has to be designed further, it decomposes into a set of related FUs from a lower order, see part of figure 2. For more detailed information on the GARM, see Gielingh [Gielingh 88]. The GARM constructs give us some advantages that will be shown later in this paper. For further detail on the information model for residential buildings, please refer to De Waard and Tolman [De Waard 91b and De Waard [De Waard 92].

Information model for building regulations

There are different ways to look at building regulations [De Waard 91a]; [De Waard 92]. Looking at building regulations from a GARM point of view, it can be said that building regulations limit the admissible technical solutions for a certain functional unit. The Building Decree gives requirements for functional units that their technical solutions must meet. The properties from the technical solution should either directly conform to the requirement or the result of an analysis, where a property can be used as input, should be conform to the requirement. This is presented in the NIAM model of figure 2.

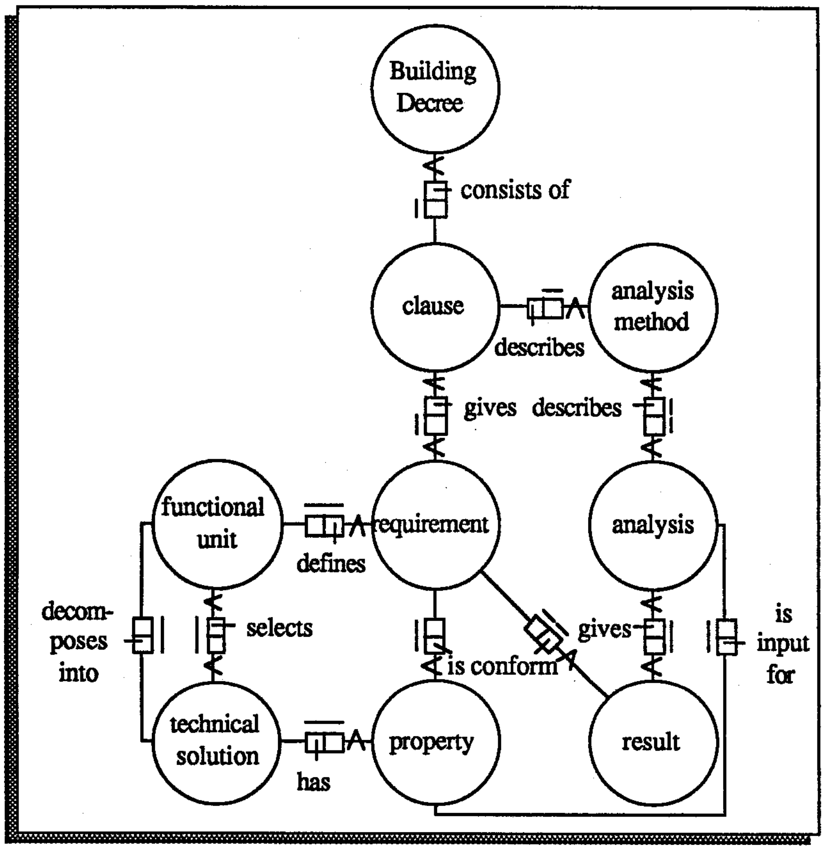

Figure 2. Relation between the Building Decree and the GARM

Seen from a database point of view, one can say that building regulations are constraints over a building model [Fenves86]. From an information point of view, it can be said that building regulations, apart from giving constraints over the building, also describe a building. This is also seen by Cornick et al. [Cornick 91] when they state: "Building regulations such as the Building Code can be separated into two conceptual entities. Models which describe the objects that are subject of the regulation and constraints, which limit the attributes and characteristics of the objects defined in the regulations. Constraints also place limits on the relationships that can hold between objects". In modeling building regulations, one should not only model the constraints, but also model the objects. Cornick et al. [Cornick 91] state: "At present, regulations only implicitly contain models of objects within their scope and subsystems. These models need to be made explicit for incorporation into intelligent design systems". The way of modeling building regulations proposed here intends to do just that. In modeling building regulations three steps are proposed:

- make a NIAM model of the entity types involved in the requirements under consideration;
- translate the NIAM model to an Express model;
- refine the Express model by adding attributes to the entity types and adding functions, procedures and extra constraints to the model.

A NIAM model can be constructed by looking at the requirements under consideration and collecting all the nouns in these requirements. Model these nouns in a NIAM model and give the interrelations between these nouns. In GARM terminology, these nouns constitute functional units, for which technical solutions should be found. The next step is to translate the NIAM model to an Express model [ISO 90]. This is done because not everything in the requirements can be modeled in NIAM. However, NIAM has a good presentation that can be used for discussion about the model. The next step is to refine the Express model by adding attributes, necessary for the determination of satisfaction or violation of the considered requirements. The last step is to add functions, procedures and extra constraints. The functions and procedures are used to describe the rules to determine whether a requirement is satisfied or violated. Some requirements can not be modeled by the use of functions and procedures. They are modeled by adding extra constraints to the model. Functions and procedures can also be used to model constraints. For an example of the way of modeling building regulations, please refer to De Waard and Tolman [De Waard 91a and De Waard [De Waard 92].

It is possible to combine the three viewpoints, the GARM viewpoint, the database viewpoint and the information viewpoint, in one model. The NIAM model, describing the entity types subject of the regulation, can be expressed in GARM concepts. The subjects of the regulations are the functional units in the model. Technical solutions for the functional units can easily be defined. The attributes that are added to the Express model after the translation of the NIAM model to Express, are properties of the technical solutions. The constraints given by the regulations are modeled by a combination of Express constraints, functions and procedures, where the latter two are mainly used for describing the analysis method defined in the requirement.

The three steps described before can be applied to each requirement in the building regulations. This way an Express model for each requirement is generated. If necessary, the models for all the requirements can be combined, so that an overall model of the building regulations develops. The model that then arises is in principle a 'simplified' product type model for residential buildings, the building regulations view on the product type model for residential buildings. However, integrating the different models of the clauses to one model, would cause unnecessary constraint complexity and would complicate model maintenance.

The building regulations can now be modeled apart from the product type model, which gives advantages as described by Fenves [Fenves 86]. Cornick et al. [Cornick 91] even state: "By separating the models from the constraints imposed by the codes we hope to provide a system which can be adapted to changes in the regulations by changing the constraints on the models, rather than having to make changes to the models themselves. Over time, the models in regulatory codes are stable and the changes that do occur are generally associated with constraints on the models".

Computer Aided Conformance Checking

In the previous sections an information model both for residential buildings as well as building regulations was described in global terms. To be able to achieve computer aided conformance checking the two information models must be related to each other. In this section the relation between the two models will be described and how that helps in achieving computer aided conformance checking.

Relating both information models

To describe the relation between the two information models we will first have a look at the three schema framework for database management systems as defined by ANSI/SPARC [ANSI/SPARC 78]. The framework proposes to distinguish three schemas for a database management system, see figure 3. The most important is the conceptual schema. It comprises a unique central description of the various information contents that may be in the database. The second important schema is the external schema or rather the external schemas. Users and application programs may view the data in a variety of ways, each described by an external schema. Each external schema is therefore derived from the common conceptual schema. The last schema is the internal schema which describes the physical storage structure. The product type model for residential buildings is a model that can be found on the level of the conceptual schema. The product type model seen from the viewpoint of the building regulations is found on the external schema level.

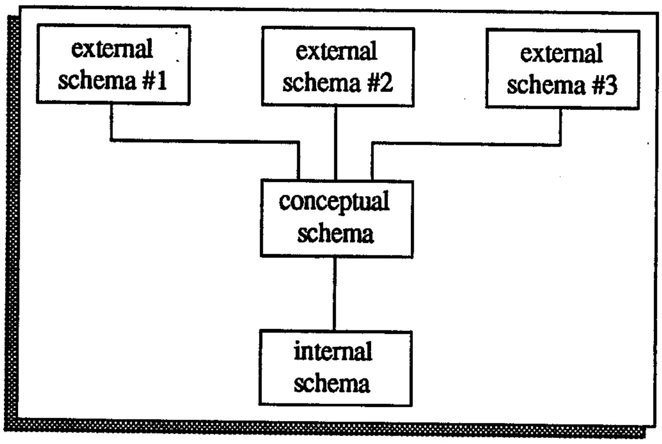

Figure 3. The three level framework for DBMSs defined by ANSI/SPARC

Seen in terms of Van Nederveen [Van Nederveen 91], it can be stated that the information model for building regulations consists of many aspect models. A clause or a combination of clauses from the Building Decree, deals with one aspect of the building. Aspects dealt with in the Building Decree are safety, usability, health and energy. If the models dealing with the same aspect are combined, different aspect models develop. Modeling a clause from the Building Decree is modeling a sub-aspect model.

A question that remains to be solved is how the two information models can be related to each other. The product type model for residential buildings is used as a view independent model, see Van Nederveen [Van Nederveen 91]. The model is described using GARM concepts. The information model for building regulations also describes a product type model for residential buildings, only this time the model is view dependent. The view is that of the building regulations. The model is also described using GARM concepts. As the product type model defined by the building regulations is a view dependent model, it must be possible to derive the view dependent model from the view independent model. The view dependent model differs from the view independent model with respect to entity types and relationships between entity types considered.

The Building Decree describes a product type model for residential buildings in more global terms than the view independent model does. An example of this is that in the view independent model entity types like 'living room' and 'bed room' are found. In the building regulations view, these spaces are both called a 'residence space', which is still a specialization of the entity type 'space' from the kernel model. For these cases a simple generalization/specialization tree can be build, where all the entity types from the view dependent model are referred to by the view independent model as in figure 4.

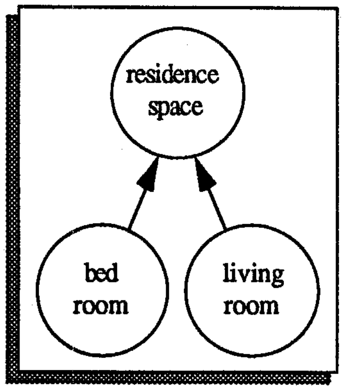

Figure 4. Example of mapping entity types from the view independent model on the view dependent model

Another change seen in the view dependent model are the relationships between entity types. Some relationships have to be redefined, because an entity type relating two other entity types to each other is not of interest in the view dependent model. Some relationships can be redefined. An example of this is i.e. a 'gas supply connection'. In the view independent model, as can be seen in the kernel depicted in figure 1, a gas supply connection is related to a 'space boundary'. However, building regulations don't state that gas supply connections should be on a certain wall, but they state that a certain space must have a gas supply connection. So the relationship between a gas supply connection and a space boundary in the view independent model, can be replaced by a relationship between a gas supply connection and a space in the building regulations view model. The relationship is however also present in the view independent model, only indirectly.

The computer aided conformance checking process

The steps in the computer aided conformance checking process are now as follows. The design team either work directly with the product type model for residential buildings, or make sure that whatever model they use can be translated to that product type model. The model is used as a neutral exchange format for exchanging information between the design team and the building authorities. The building authorities receive the model in the neutral format. The neutral format is translated to the building regulations view on the neutral format. Within the view dependent model, the constraints as defined by the building regulations are found as constraints, functions and procedures. Each requirement is now checked for applicability to the design at hand. If the requirement is found applicable a consistency check is performed. If the model is found to be consistent, the requirement is satisfied, otherwise the requirement is violated. Doing this for all the requirements in the building regulations comprises a conformance check of a building design against building regulations.

Implementing Computer Aided Conformance Checking

So far computer aided conformance checking has been discussed from a theoretical, conceptual point of view. In this section an environment will be sketched with which it is possible to achieve computer aided conformance checking based on the models described earlier in this paper.

To accomplish computer aided conformance checking four steps need to be performed (figure 5). The first is the creation of a product model according to the structure of the product type model. The output of this step is a filled product model database, containing the description of a design that is to be checked for code conformance. Next to that, the regulation model must be implemented. The output of this step is an empty regulation database, containing the entity definitions as given in the entity type model of the model of the regulations. Further output is source code that can check a filled regulation database for code conformance based on the constraints, functions and procedures defined in the constraint part of the regulation model. The third step is filling the regulation database by extracting the necessary information from the filled product model database. Once the regulation database is filled the source code is used to perform the conformance checking process which results in a conformance report.

Implementing the product type model for residential buildings

The product type model for residential buildings is implemented in a layered way [Willems 91]. To be able to make a product model for a residential building based on the product type model at least three layers are distinguished. These are the reference model layer, the product type model layer and the product model layer, see figure 6. On the reference model layer models like the GARM and the kernel of the product type model can be found. Also other abstract models useful for developing product type models could be found here. The product type model layer inherits from the reference model layer, it uses the reference models described in the reference model layer. Entity types from the product type model are specializations of the entity types defined in the different reference models where the inheritance structure is sometimes a multiple inheritance structure. Entity types inherit at least from the GARM and possibly also from other reference models such as the kernel. On the basis of the product type model, defined on the product type model layer, a product model can be made using the structure defined in the product type model. The product model is an instantiation of the product type model. It is not always clear what can be found on which layer. An example of this is the kernel for the product type model. The kernel can be placed on the reference model layer, as is done in figure 6, but the kernel can also be modeled as a sub-layer of the product type model layer. However, it is not important on which layer it is defined. Even a number of extra layers can be defined more abstract than the reference model layer, as proposed by Willems et al. [Willems 91].

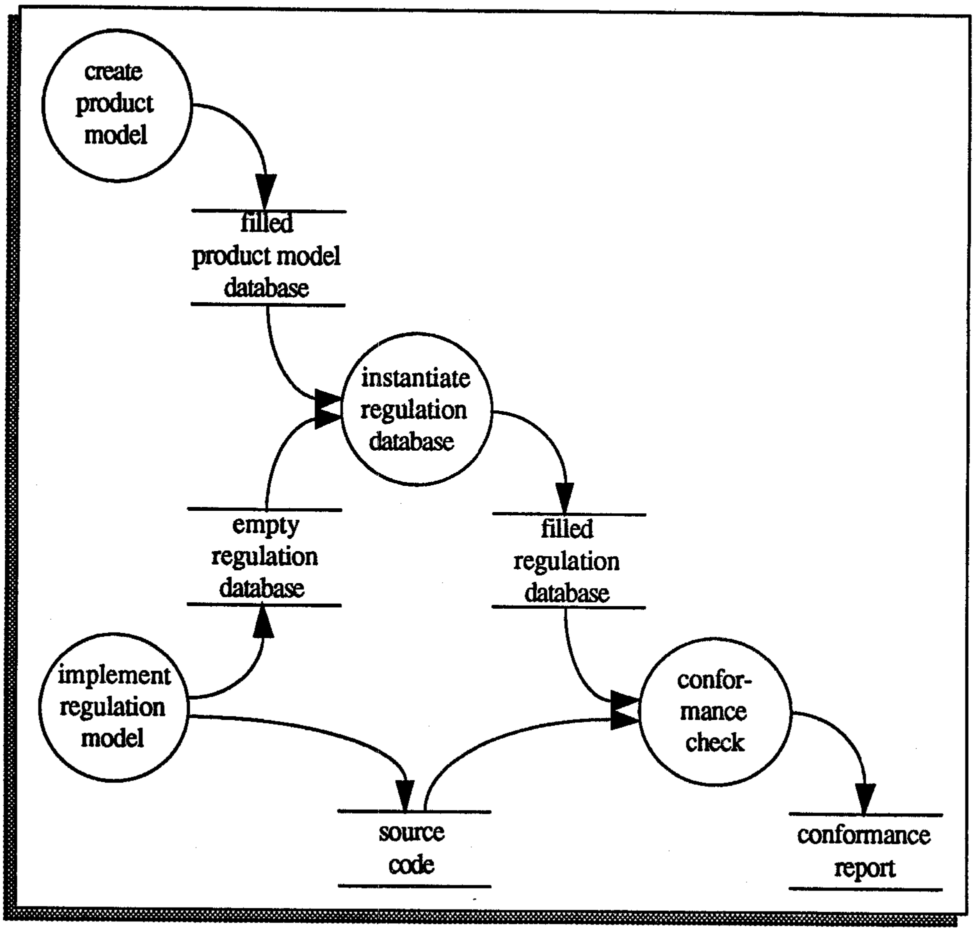

Figure 5. Data flow diagram for a Computer Aided Conformance Checking Environment

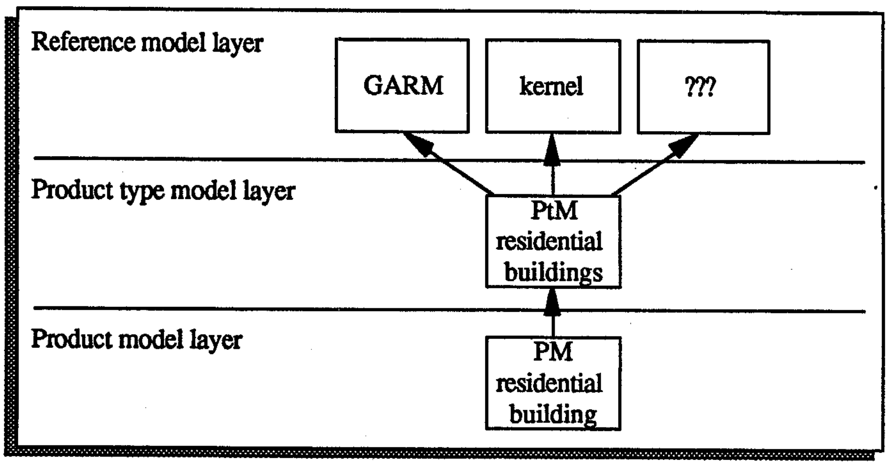

Figure 6. The layered structure of the product type model for residential buildings

For the implementation of this multi layered structure software has been developed at TNO Building and Construction Research, called PMshell (Product Modeling shell) implemented in the object oriented language Eiffel [Meyer 88]. PMshell works directly on a database. Based on the database the software is able to generate Eiffel class templates, which can be further refined towards an actual implementation. The product type model described in Express can now be translated to an Eiffel implementation in the following way. An Express parser directly generates an empty database. Because of the layered structure this is done in multiple steps. First the models on the reference model layer are parsed, because the product type models are based on these models and they have to be known before a product type model can be made. The next step is to parse the Express models on the product type layer. The database is now filled with entity types from the product type model. On the basis of the database Eiffel class templates can be generated automatically. This is the first step towards the implementation of a product modeller for residential buildings. By adding functions and procedures to the Eiffel classes a product modeller is constructed. The product modeler is then used to fill the last layer, the product model layer. With the product modeler an end-user is able to instantiate the product type model and as such define a product model.

Implementing the model for building regulations

To implement the building regulations model, the Express model of the building regulations has to be processed. For the processing of the Express models, software developed in the cause of the Esprit II project called IMPPACT is used [Van Mieghem 91]. The Express software developed there consists of three components: (1) an Express parser, (2) an Express semantic checker and (3) a rule and procedure generator. The first two components are there to see that the Express models are valid models according to the definition of Express. The last component generates source code for the calculation of derived attributes, source code for performing functions and procedures and source code to determine whether rules defined in the Express model are violated or not. Based on these components and software developed during the research a building regulation model is implemented. Based on the Express model of the building regulations under consideration an empty database is generated containing all the entity types and attributes defined in the view model of the building regulations. Next to that the source code as mentioned before is being generated so that each instance of an entity type can be checked for conformance with the defined rules.

Checking for conformance

To be able to perform computer aided conformance checking, the empty regulations database is filled from the filled product model database. After the database has been filled, the source code generated is able to check if the information contained in the database is conform the regulations considered at that particular moment.

Based on the strategy described in the previous sections implementations were made for two clauses from the Dutch Building Decree. With the information systems that were developed, it is possible to check a residential building for conformance with the two clauses automatically [De Waard 92].

Advantages and New Possibilities

In the previous sections, two information models were discussed and how these two information models can be related to each other, where an entity type model of a clause is a view of the building regulations on the product type model. By separating the two models and relating them afterwards, we have gained some advantages as also described by Fenves and Garrett [Fenves 86]:

- changes in the regulations produce no change in the product type model and vice versa. Only the changed model of the regulation, and possibly the mapping program requires modification. However, if Cornick et al. (1991) are correct in suggesting that the constraints are the most unstable of the regulations, the mapping program may not even have to be changed, as it only maps entity types onto each other.

- the software can be made applicable to any regulation or standard which can be modelled in the way described in section 3. The model of a building regulation must be replaced by that of an other. The mapping program should then also be changed.

The above advantage strongly gains in importance as the realization of a common European market rapidly approaches. However, before models for building regulations can be applied across national borders, the international community will have to agree upon a structure for modeling buildings, such as a product type model for residential buildings. If this can be accomplished, only one mapping program per model of a building regulation will have to be developed to fill the empty regulation database. Otherwise, mappings will have to be written for each combination of a building model and a building regulation model.

As stated before, the different models for the clauses can be integrated into one overall model. However, not integrating the models results in more simple and better maintainable models. Not integrating the models also gives an other advantage. Now, for each clause a separate system can be made that checks for code conformance. This opens the way for simultaneous checking of designs.

By describing the way of developing an information model for building regulations, a new opportunity has been created: information models for building regulations can now be developed by the authors of these regulations, undoubtedly the persons most suited for the job, after training. This greatly eliminates the risks of personal (mis)interpretation by whoever is modeling the building regulations.

By achieving computer aided conformance checking it is also possible to do something about other building regulations related issues. First of all, it might finally become possible to see how all kinds of regulations, issued by different authorities, are interfering with, if not neutralizing, each other. With the fast amount of regulations it is almost impossible to do a job like that by hand. It is clear that when the Ministry of Traffic states that houses should be build in the close vicinity of railway stations because of the gain in mobility of the people, and the Ministry of Housing states that houses should not be build in the close vicinity because of safety risks, that these two requirements neutralize each other. But there might be much more subtle interference between different requirements that are not immediately clear and detected. Another possibility opened with the achievement of computer aided conformance checking is, that it becomes possible to study the effect of the introduction of new requirements and standards. This can be reached by selecting a number of houses and simulate the conformance checking process to see what the effect will be on the houses.

Epilogue

In this paper it was shown that we can actually achieve computer aided conformance checking when the theories developed with regard to product models are taken as a starting point. In the paper we showed how product models can be used to introduce architectural semantics in models describing buildings. The model considers two ways of looking at buildings and integrates those two views into one model. As to the model for building regulations we can state that it consists of two components. A model of objects in the regulations together with the interrelationships and the constraints defining the constraints given in the regulations. The model of the objects is at least as important as the constraints, as the model of the objects describe the view of the regulations considered on the building. To achieve computer aided conformance checking we 'merely' have to map the product model on the object model of the building regulations. With the available software and software developed a building can be checked for conformance with building regulations with the aid of computers as was shown in the course of the research. It is my opinion that the method described is not limited to the housing industry but also for other industries.

However, there are also some questions that still need answering. The research mainly dealt with computer aided conformance checking in a passive sense, where there is a finished design that is checked for conformance with the building regulations. But what is with the active use, where regulations are used (inter)actively during the design process. Does the way of modeling the regulations remain the same and does the way of interaction with the users changes or do we have to look for another way of modeling the regulations? In my opinion the first solution is the most likely. Another issue that should be mentioned is the change in the nature of regulations. Regulations considered during the research were of a technical nature. However, more and more we see functional regulations. The question that needs answering is how to model those regulations, although it is my opinion that in modelling these regulations the emphasis will lay more on the model of the objects than on the constraints. And then of course, there is the question how to support the writers of regulations. Can they be supported with the models described in this paper?

So, some questions have been answered in the research but there are still many issues that need looking into.

References

[ANSI/X3/SPARC78] ANSI/X3/SPARC Study Group, Database Management Systems, *Framework Report on Database Management Systems*, AFIPS Press, Montvale, NJ, 1978

[Cornick 91] Cornick, S.M., Leishman, D.A., Thomas, J.R., "Integrating Building Codes into Design Systems". In: Kähkönen, K., Björk, B-C., *Proceedings of the International Workshop on Computers and Building Regulations*, 27-29 May, 1991, Espoo, Finland.

[De Waard 90] De Waard, M., Tolman, F.P., "Implementing the Building and Regulations View on Buildings". In: Wagter, H., *Proceedings of the Second CIB W78 & W74 Joint Seminar on Computer Integrated Construction*, 17-19 September, 1990, Tokyo, Japan.

[De Waard 91a] De Waard, M., Tolman, F.P., "Modelling of Building Regulations". In: Kähkönen, K., Björk, B-C., *Proceedings of the International Workshop on Computers and Building Regulations*, 27-29 May, 1991a, Espoo, Finland.

[De Waard 91b] De Waard, M., Tolman, F.P., "A Model to Describe Residential Buildings". In: *Proceedings on the Second International Workshop on*

Computer Building Representation for Integration, 3-5 June, 1991b, Aix-les-Bains, France.

[De Waard 92] De Waard, M., *Computer Aided Conformance Checking: checking residential building designs against building regulations with the aid of computers*, Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, March 1992.

[Eastman 89] Eastman, C.M., "Architectural CAD: a Ten Year Assessment of the State of the Art". In: *Computer Aided Design*, Vol. 21, No. 5 (June 1989), p. 27-36.

[Fenves 86] Fenves, S.J., Garrett, J.H. Jr., "Knowledge Based Standards Processing". In: *Artificial Intelligence*, Vol. 1, No. 1, 1986, p. 3-14.

[Gielingh 88] Gielingh, W.F., *General AEC Reference Model (GARM)*, BI-88-150, TNO Building and Construction Research, Rijswijk, October 1988.

[ISO 90] ISO, *Exchange of Product Model Data - Part 11: The Express Language*, Document CD 10303 - 11, ISO TC184/ SC4 N64, July 1990.

[Meyer 88] Meyer, B., *Object Oriented Software Construction*, Prentice Hall, 1988.

[Nijssen 89] Nijssen, G.M., Halpin, T.A., *Conceptual Schema and Relational Database Design, a fact oriented approach*, Prentice Hall, Sydney, Australia, 1989.

[Van Mieghem 91] Van Mieghem, R.B.P., De Bruijn, W.J., *ISP Reference Manual*, ESPRIT II - CIM III.1.3.1, #D.WP4-CEC.4100/005, June 1991.

[Van Nederveen 91] Van Nederveen, G.A., Tolman, F.P., "Modelling Multiple Views on Buildings". In: Wagter, H., *Proceedings of the CIB W78 Seminar "The Computer Integrated Future"*, 16-17 September, 1991, Eindhoven, The Netherlands.

[Willems 91] Willems, P.H., Kuiper, P., Luiten, G.T., Luijten, B.F.M., Tolman, F.P., "A Framework for Evolutionary Information Model Development". In: Wagter, H., *Proceedings of the CIB W78 Seminar "The Computer Integrated Future"*, 16-17 September, 1991, Eindhoven, The Netherlands.