

THEORY AND IMPLEMENTATION OF A CONCEPT MODELER SUPPORTING THE HVAC ENGINEER

Ir. H. Schevers and Prof. Ir. F.P. Tolman
Delft University of Technology, Faculty of Civil Engineering and Geosciences
Stevinweg 1, P.O. Box 5028, 2600 GA Delft, the Netherlands

ABSTRACT

In a larger research program on 'cost versus value evaluations in the early design stages of technical buildings', a study and software implementation has been made to simulate HVAC expertise using an Integrated Early Design Environment. Basically the idea is to create a virtual early design software environment where the development of the requirements, concept designs and evaluations on costs and values is supported. A prototype implementation has been made that uses Product Data Technology (PDT) in combination with Knowledge Technology (KT) and which is able to calculate certain HVAC aspects. This paper describes how an existing HVAC simulation program has been used to support the design process in the inception and early design phases.

INTRODUCTION

Product modeling or Product Data Technology (PDT) started in the early 80's. The basic idea was to exchange and share data about building artifacts based on open neutral information models. In the mid 80's ISO STEP [STEP] (Standard for the Exchange of Product model data) started to develop a standard for such models. The initial aim of STEP AEC (Architecture, Engineering and Construction) was to exchange product data between different CAD systems. Therefore a great effort was made in the development of models for shape representations. Later, an industry-wide standardization effort was initiated by the International Alliance for Interoperability (IAI). They created the Industry Foundation Classes [IFC], a product model that is supported by CAD vendors like AutoDesk, MicroStation etc. These product models are particularly focusing on the design phase. This paper describes a research project at the Netherlands Organization for Applied Scientific Research (TNO), department Building and Construction Research (TNO-Bouw) that uses the PDT approach to model the *early* project phases and especially the inception phase and early design phase. The aim is not to focus on data exchange but rather to focus on an integrated design environment. This software environment

supports the development of requirements, concept designs and evaluations on costs and values. With such an environment, different scenario's can be examined. This means that different sets of requirements and/or different designs can be examined easily. The design process is supported with automatic evaluations and suggestions regarding the requirements and concept design. Using for example HVAC software in the early design phase can result in a design with good HVAC properties. In practice, HVAC expertise is often only consulted in the detailed design process. The influence, the HVAC expertise has on the design itself, is not integral and mostly limited to only the HVAC installations. Sometimes extra installations are needed to meet the requirements while taking HVAC aspects into account in the early design process could prevent this! This paper describes an integrated design environment that supports some HVAC expertise.

ANALYSIS

Decisions in early project phases of large-scale engineering projects such as power plants, process plants, hospital facilities, offices, civil engineering and infrastructure, have a large influence on the course of these projects. The direction of the project is directly dependent of these decisions (see figure 1)

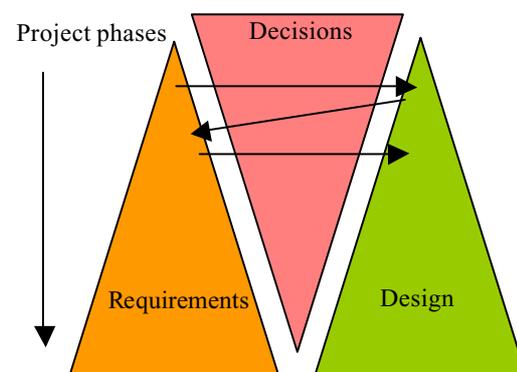


Figure 1 - development of the requirements and the design in combination with the impact of design decisions-

When an office organization for instance has grown out of its housing, possibilities like building a new housing accommodation or enlarging the current one, are available. The decision what solution to pursue is often made at an early stage in the project. Usually these early decisions are based on poor and insufficient information. A lot of information is based on experiences and assumptions, which are not always correct. This can result in incorrect decisions, which often results in extra costs. Furthermore the complexity and uniqueness of large-scale engineering projects make it difficult to oversee the whole project:

- Projects in crowded environments have many governmental rules and restrictions. These rules and restrictions are present to protect the existing situation. Sometimes different authorities are present and guard their own interests. Usually neither all restrictions nor empowered authorities are present at the early stages. Determining if the early decisions will lead to situations where restrictions are applicable is difficult.
- In case of a tendering process, the available time is mostly limited. Aspect evaluations supporting decision processes are not always ready or available, especially when changes are made until the last minute.

- When a project is complex, evaluations can be too simple and thus incorrect. Even lack of information or lack of relevant criteria can result in bad design decisions. In the early phases of complex projects it is often not apparent which properties are relevant.
- Re-use of knowledge gathered in previous projects is not stored in a neutral format and therefore not easy accessible.

Supporting the early design phases in such a way that the quality of the decisions improves is beneficial for the whole project. A lot of research in this area is carried out by projects like CONCUR, Seed and CoBrITe. The development of the requirements, concept designs can be supported by (re-) using knowledge from previous projects or using existing software programs for evaluation purposes.

SOLUTION CONCEPT

Basically the idea is to apply the principles of Product Data Technology (PDT) integrated with knowledge technology (KT). With Virtual Reality (VR) based GUI's, a virtual project environment can be created. Using this environment, it is possible to explore the boundaries of the project by developing requirements and concept solutions on trial and error basis.

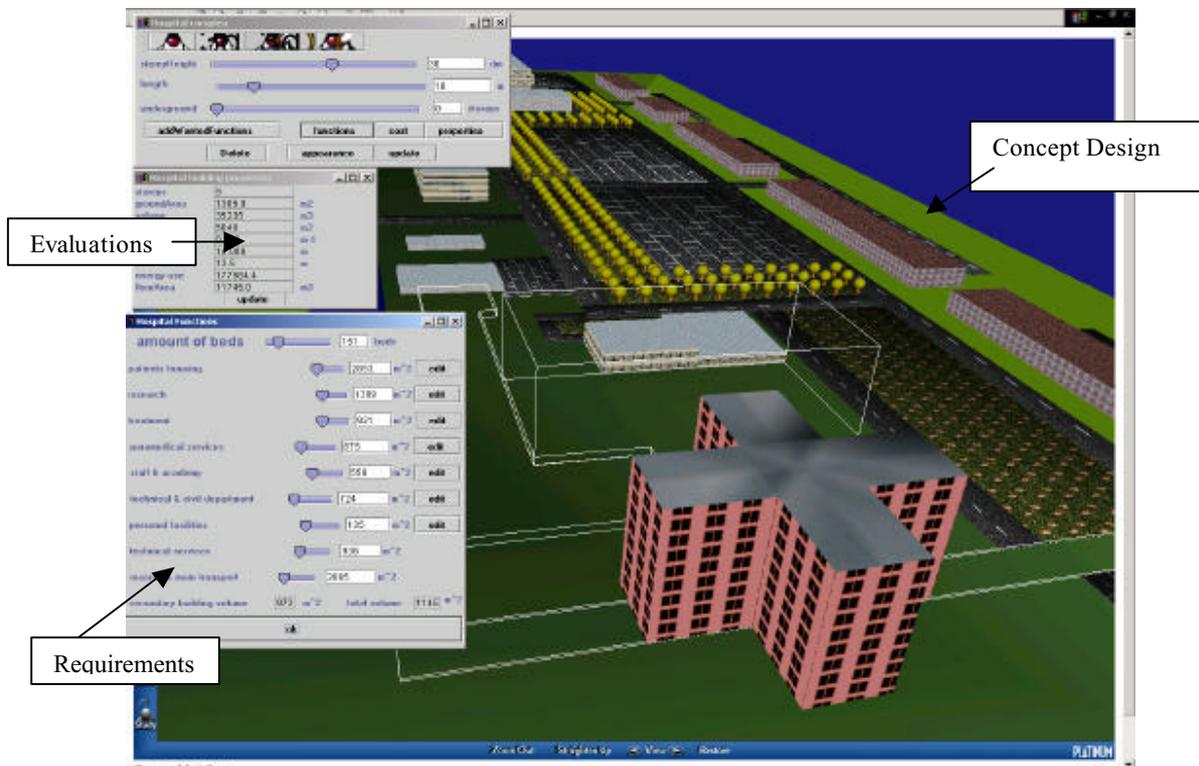


Figure 2- screenshot: Virtual Early Design Environment for Hospitals-

The values of the concept solutions can be tested by multi aspect evaluations. Receiving support with developing the requirements and the concept design in combination with direct feedback on multi aspects, actors within the project can orientate well and make the correct decisions for the real project. To ensure more insight on this solution concept, a prototype implementation can be introduced to create a better view of such an environment (see figure 2). This prototype does not deal with the real problems of creating such an environment but gives a good idea of how the environment should work [Schevers & Tolman 2000]. In this prototype it is possible to model existing hospital situations and create a concept solution by demolishing existing buildings and creating new ones. By implementing some knowledge about hospitals, the size of the new buildings can be calculated in order to meet the given requirements. Furthermore knowledge about costs is available which will result in a direct cost feedback.

METHODOLOGY: PRODUCT DATA TECHNOLOGY

Product Data Technology (PDT) is a technology to describe products in a computer interpretable language. The idea is, like all other models, to describe the relevant properties of the product only. In that case the model becomes an abstraction of reality.

The product model, which is the result from this research, should describe a building including all relevant aspects for the early project phases (inception and early design phase). This means the product model must contain objects and properties regarding structural engineering, thermo related properties and all other relevant disciplines. All properties regarding the building's lifecycle must be integrated into the model as well. Because civil engineering projects have an unique character in combination with the iterative early design process, it is difficult to create one product model which fits every project and which is capable to capture all relevant project data regarding the iterative early design process. This means the product model must be able to cope with the development of the requirements and the development of the concept design (see figure 3). Furthermore the product model must be extendable in order to cope with unique situations. The focus of this project is to create such a product model that supports the early phases of a building project and which at least takes into account the HVAC aspects.

METHODOLOGY: OPEN PRODUCT DATA TECHNOLOGY IN THE EARLY PHASES

The range of available information of the early phases of a project often starts from virtually nothing till a concept design with an initial list of requirements. More and more information is gathered

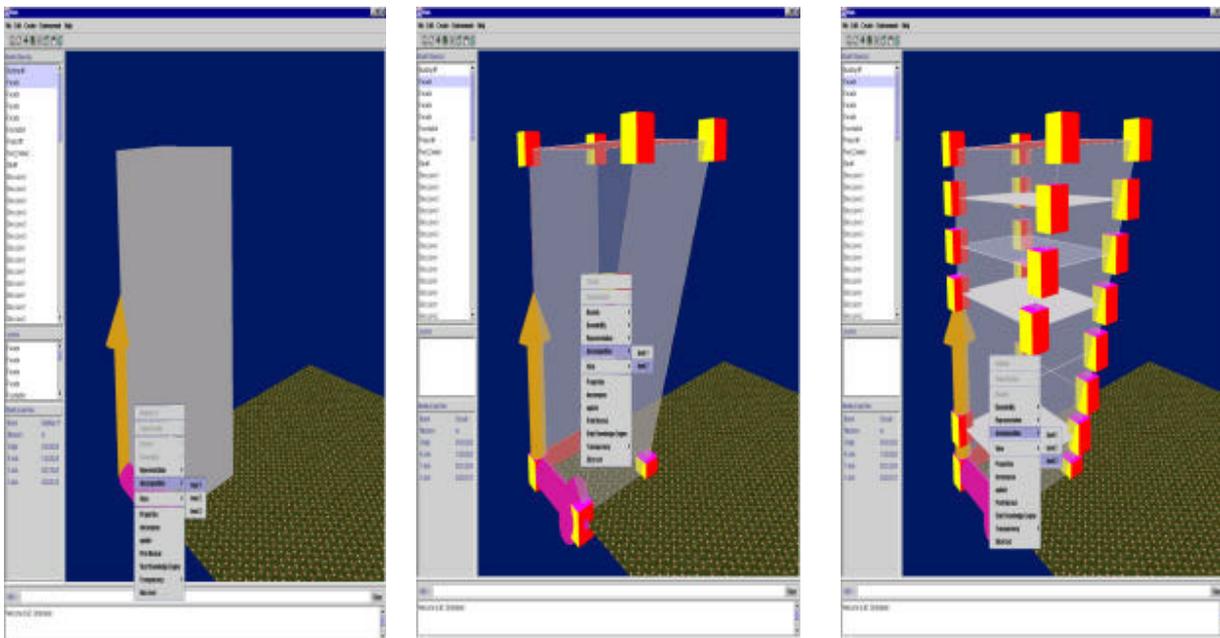


Figure 3- screenshots: development of a Building in three different levels of detail-

and processed during the project. This information may contain information regarding requirements or design information. The client can refer to concepts (s)he is familiar with. In the case of a hospital project the client can refer to other existing hospitals. Also, evaluation studies have to be made in order to determine the feasibility of the project. In an iterative process of continuous refining, requirements are developed in combination with the design. Basically that means that the amount of information keeps growing and becomes more accurate over time (see figure 3). Supporting this process means that the data model must be able to follow this process. The product model must support the growing in levels of detail (LOD), accuracy, etc. Furthermore the semantic definitions and relations can differ in every project. Creating a product model with semantic objects including topologic relations between these objects can result in a rigid model. Changing the model at 'run-time' by adding new object types gives more flexibility to model the early design phases. Integrating such a model with knowledge in order to support the design process is more difficult compared to a rigid model. In a new prototype two mechanisms are implemented to give the product model more flexibility. The first one is to add new types in the schema of the product model. The second one is to dynamically add extra properties into existing objects. Supporting LOD's without specifying the different levels in advance gives the model enough openness to capture all the project information. From an information modeling point-of-view the main question of such an approach is: *how to keep this model consistent and true?* It is necessary to make sure that the model does not become a chaos of objects, therefore some relations have to be predefined. Connecting the objects with these predefined relations means the objects will respond to each other. How the object must respond is mostly in the property level of the object. To deal with the desirable responds of the model 'behaviours objects' are introduced. These 'behaviours objects' respond to a certain product data structure in the product model. For example a 'derived property behaviour object' has been implemented which responds when an object with a property is being decomposed into other objects. At that moment the property becomes derived from the properties of the composition objects. The 'behaviour object' searches those composition objects in order to determine whether its property is derived from the composition properties or not. The 'behaviour object' uses the semantics of the property to see if such behaviour is desirable. With this approach the model can be kept consistent and is flexible enough to model relevantly.

EXPERIMENT: HVAC SUPPORT IN THE EARLY PHASES

To support HVAC aspects within an integrated early design environment, basically two approaches are feasible. The first is to abstract the HVAC knowledge and express it in simple rules usable for product models, which are designed for the early project

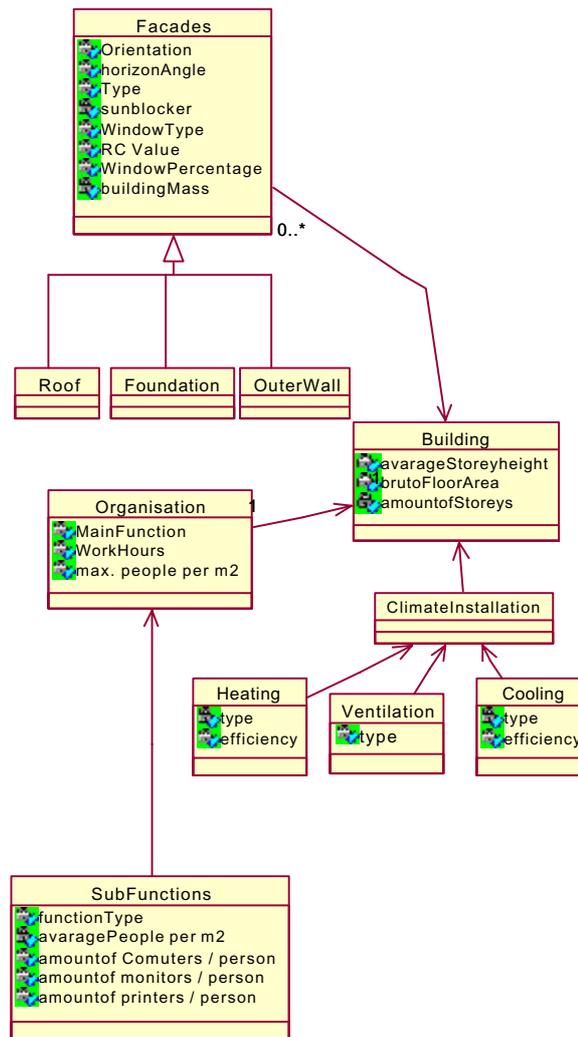


Figure 4- the HVAC model: identification of important objects and properties in order to simulate the energy consumption

phases [Tolman et all 1999]. The second approach is to use existing HVAC engineering software and to apply building knowledge to generate a reasonable detailed building concept from the initial data

available. The advantage of the second approach is of course that the generation of a reasonable detailed building concept can be guided by the client's requirements. Secondly the actual HVAC knowledge is better represented in these existing applications than in a set of abstracted knowledge rules. This approach has been adopted in this study. An existing program called VA114 (www.vabi.nl) can be used to simulate climate energy consumption. This program is able to calculate temperatures, the amount of hours when a certain temperature has been exceeded, energy need for temperature control, etc. The program is able to take several installations into account and is based on a multi room calculation where interaction is modeled between the rooms. Furthermore shadows of the building itself or shadows of other buildings are also taken into account. The necessary input for this program consists of detailed design information. Starting from the idea that this information is not available and hard to automatically generate or derive, the most important objects and properties for making an assumption of the energy consumption are identified. In the next diagram (figure 4), objects with the properties and their relations are displayed. The modeling language is the Unified Modeling Language (UML). What you see in figure 4 are object classes connected with two types of relations: bi-directional relations (with cardinalities and roles), specialization (with closed arrows). Each class may contain attributes. The reason for using UML is that the class diagram is used as product model representation and that it is possible to generate Java code out of this class diagram. The static product model is augmented with extra default values to provide the detailed information needed for VA114. This means default values and simple knowledge rules are used in combination with this HVAC model to generate the detailed input needed for the calculations. By identifying the key objects and properties the influence of the necessary detailed values can be neglected compared to the accuracy of the available information. Within the model several enumerations have to be set. For example in the 'SubFunctions' object 'functiontype' is an enumeration of: office, store, meeting room, reception, computer room, technical room, transport, washroom, etc. Also for the installations several enumerations are defined like for cooling and heating: water, cooling, combination and for ventilation: axial ventilation, centrifugal ventilation, etc. With this model it is possible to calculate the climate electricity energy consumption of the ventilators, heating and cooling, gas usage and the temperature within the building.

METHODOLOGY: MAPPING FROM AN OPEN MODEL TO A STATIC MODEL

With the open product model approach a building model can be created that is flexible enough to capture relevant project information. Using the 'behaviour objects' the model is kept consistent. Also the relevant HVAC objects and properties are identified for calculating certain HVAC aspects [HVAC model] (see figure 4). The next step is to make sure all the necessary information within the open product model is available to produce the HVAC model in order to calculate the HVAC aspects. Because the open product model is very flexible, it is easy to assert new properties that are important for the HVAC model. Using the schema of the HVAC model, the system asserts the needed properties to the correct building objects, which are available, assuring all information in the static HVAC model is also available in the open product model.

In a particular case, a Building object exists but does not have relations with objects like walls, the building envelope or the foundation (which are needed for the HVAC model), a knowledge system takes over and generates these objects. This system can be used to generate the objects needed for the HVAC model automatically. For example figure 2 shows three different levels of detail in a building. These steps can be made automatically by the knowledge system or manually. This assures that all objects needed for the HVAC model can be added automatically. Adding the needed properties to the model, the HVAC model can be produced. If mapping is needed between objects or between properties, new 'behaviour objects' can be created to store the knowledge for these mappings. Now the open product model can be used in combination with the knowledge base to generate the HVAC model. Of course not all situations can be entered into the knowledge base. *This means that this approach does not guarantee that in every situation a HVAC model can be generated, but it gives an opportunity to model with more flexibility than only the HVAC model itself.* Having more knowledge in the knowledge base will also enlarge the flexibility of the open model. For example, instantiating only one building object does not provide enough information to generate the HVAC model. In order to be able to predict HVAC aspects even in this situation, knowledge is used to supply the building knowledge with the extra information. The system makes the design decisions based on knowledge rules and default values in order to supply the current object

with extra information. The purpose of making these design decisions is at first not to aid the user in this process but to assure enough information is available to generate the HVAC model. With this process even while instantiating only one building object, the user can evaluate certain HVAC aspects. When the user supplies more information on the building it self, the system has to make fewer default decisions and thus the input for the evaluations is better and thus the evaluation itself is more accurate.

This means the HVAC evaluation follows the Level Of Detail of the project to a certain extent. However, big research questions are still open. But with the current implemented mechanism, HVAC evaluations are possible and the outcomes of the evaluations are more accurate when more information is available.

EXPERIMENT: PROTOTYPE IMPLEMENTATION

(This section describes the prototype implementation, which is available on <http://cti030.citg.tudelft.nl>)

A prototype implementation has been made aiming at office buildings because a lot of information on office buildings is easy accessible. In large projects dealing with office buildings, the clients requirements starts with an organization structure followed by the space requirements. The Prototype implementation provides some basic functionality to capture these organization structures and the needed spaces (see figure 5). Choosing from a library of requirements (library window in figure 5) the user is able to create a specific set of requirements (project requirements window in figure 5). Being able to add new user-defined objects to the library, the flexibility to model becomes high. Within the library, most defined objects have default properties. In figure 5 a property frame is displaying the default properties of the 'OfficeOrganisation Object'. The Requirements are ordered using a template or a view of the project data. At this moment, the requirements are viewed following a standard clients requirements list. This means first the organization is described followed by the functional spaces and at the end the design itself.

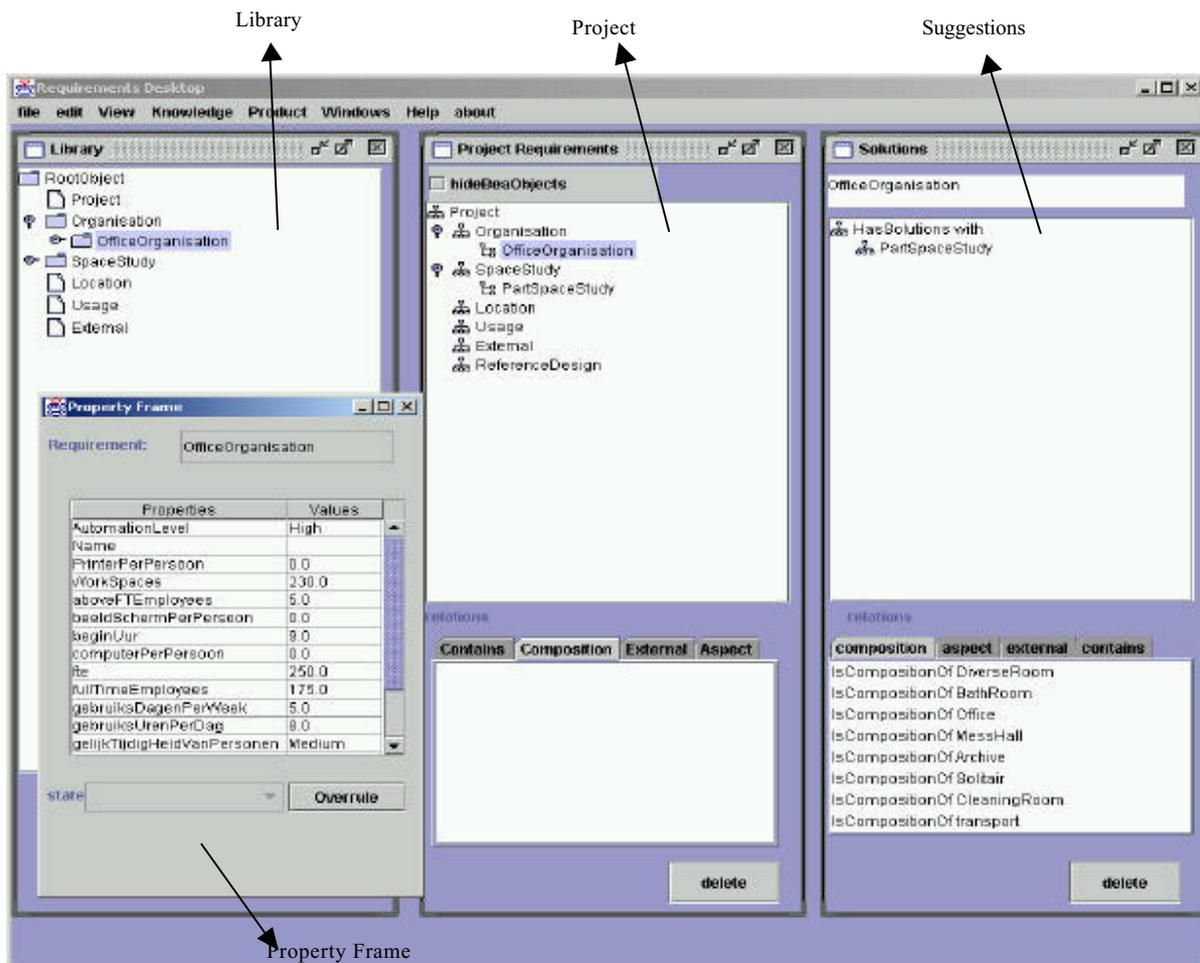


Figure 5- screenshot requirements model-

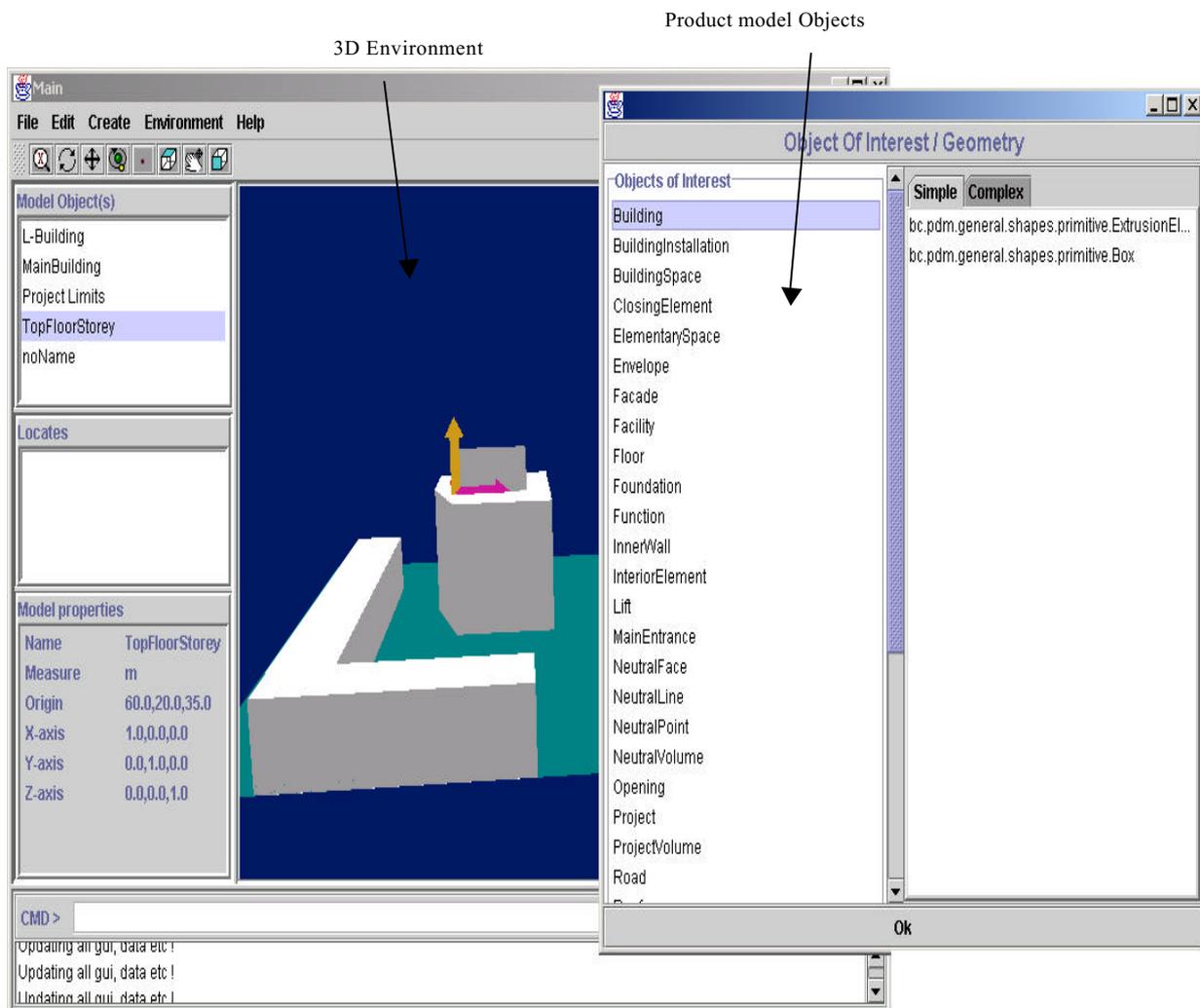


Figure 6- concept design desktop-

In the current implementation all other requirement types can be attached to the functional spaces or the design itself. For displaying the concept design a VR environment desktop has been implemented (figure 6). Within the 3d environment of this desktop, objects like building, wall, installation, parking area, etc can be instantiated with a certain shape. The shape is visualized and the properties of the objects can be displayed (see figure 6). Connecting the 'Spacestudy objects' in figure 5 to a building object in figure 6 by creating a relation between these objects, the system automatically adjusts the size of the building object to match the 'Spacestudy object'. Now it is possible to select a building in the VR design desktop and calculate the HVAC aspects. The system will identify that *not* all information is available and will add the needed information to the project.

The buildings on figure 6 will automatically be decomposed into walls, floors, and roofs, which contain the required properties (see figure 3). In this case the required properties will have a default value. Now the system *does* contain enough information to calculate the HVAC aspects. The information regarding the building will be extended automatically using the HVAC model (see figure 4). As result of the HVAC calculation the system will display figure 7. Changing the properties, the shape of the buildings or changing the newly generated walls or roofs, will have of course a different HVAC result. The results will also suggest a certain installation for the building. These installations can be added to the buildings.

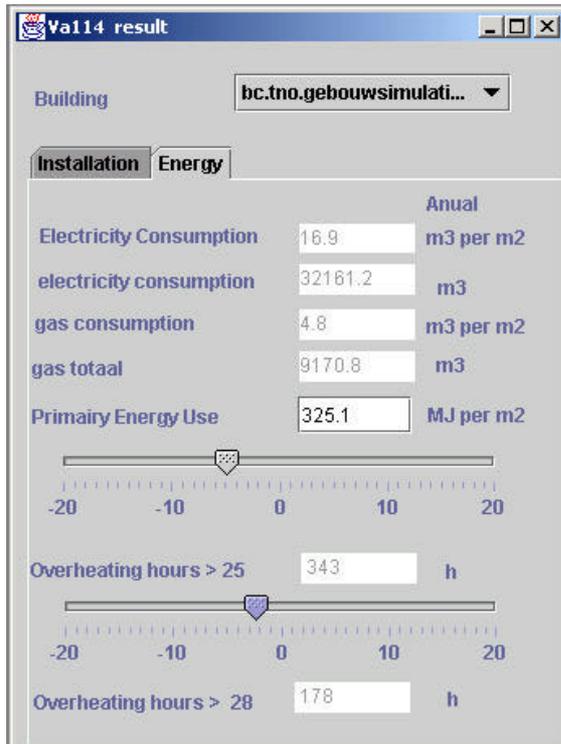


Figure 7-HVAC result frame. Electricity usage is expressed as the equivalent gas usage -

CONCLUSIONS

The open model approach offers a lot of freedom in modeling the early design phases of building projects. Adding new semantics and properties to the model during the project make it possible to capture relevant project data. Using this open model approach, a connection has been made to a static HVAC product model. This static product model is an identification of key objects and properties for some HVAC aspects. The gap between the open model and the static model is bridged using 'Behaviour Objects'. These 'Behaviour objects' have the capability to supply extra information to the open model in order to generate the HVAC model. A proof of concept implementation of this approach has been made using a HVAC simulation program. The implementation supports the capturing of project information in different levels of detail that goes parallel with the design process itself. This means the evaluation of the HVAC aspects can be done in these different levels. The accuracy of the HVAC evaluation grows equally with the project. Furthermore the interactive VR environment and the direct responds of the system make it possible to see directly the consequences of different design decisions.

The results have been very satisfactory. The reasons are that the generation process can be guided by the user requirements, i.e. if the client wants a certain comfort level for its workers, the system generates an aspect model that reflects the consequences of this requirement. Another reason is that the domain knowledge is much better represented in an existing HVAC evaluation application than in a set of abstracted knowledge rules. The next step in this research is to study the consequences of adding a second and third knowledge domain. There is little doubt that the same approach can be used in other areas. What is less certain is the question: How do these different knowledge domains overlap and how does that reflect in the outcome of the various evaluations? And if there are conflicting situations (which there sure are) how can these be dealt with? In order to make sure these techniques can be valuable, more research is necessary.

REFERENCES

- [STEP] Standard Exchange of Product data (STEP), <http://cic.vtt.fi/links/step.html>
- [IFC] International Alliance for Interoperability (IAI), <http://iaiweb.lbl.gov/>
- [CONCUR] Concurrent Design and Engineering in Building and Civil Engineering (CONCUR), <http://cic.vtt.fi/projects/concur/index.html>
- [COBrITe] Construction Briefing Information Technology. <http://www-staff.lboro.ac.uk/~cvmahh/cobrite-intro.htm>
- [SEED] Software Environment to support the Early phases in building Design. <http://seed.edrc.cmu.edu>
- [Schevers & Tolman 2000] Schevers, H. and Tolman, F.P., "Supporting the Inception Stage of Building Projects with Instant Value versus Cost Evaluations" Proceedings of the CIB78 Conference. , 2000.
- [Tolman et all 1999] Tolman F. P., Ozsariyildiz S., Schevers H., Computer Aided Inception of Large-Scale-Engineering Projects: Evaluating Value Versus Cost, The International Journal of Construction Information Technology, Volume 7 #2, University of Salford, 1999.