SUPPORTING THE INCEPTION STAGE OF BUILDING PROJECTS WITH REAL-TIME VALUE VERSUS COST EVALUATIONS

Hans Schevers, Frits Tolman

Technical University of Delft, Faculty of Civil Engineering and Geosciences

ABSTRACT: The paper discusses the ideas behind an inception support modeller that uses state of the art Product Data Technology (PDT) and Knowledge Technology (KT). The modeller under development provides facility owners, facility owners to-be and project developers with the opportunity to create and evaluate a number of alternative solutions for their accommodation by stating their requirements such as functional requirements and resources such as available money and location. By applying knowledge rules, case rules and default values information gaps will be closed in order to generate more detailed alternatives. This process assures the availability of just enough product information to perform an evaluation of the performances and costs, and gives the facility owner a better view on a realistic solution for his requirements. Using a 3D front-end in combination with a requirements language which is easy to comprehend, the client can put in his requirements on different aspects of the project, like (cash flow, type of contract), building environment related aspects (availability of public transport or green), functional related aspects and components related aspects. Feedback will be given like cost evaluations, construction time and performance values expressed in money for energy usage, durability, maintainability, walking distances and such. The working of the tool is illustrated by a case, a Hospital Inception Modeller. The case evaluates the feasibility of the renovation of a Hospital Complex in Delft in The Netherlands

KEYWORDS: Inception Support, PDT, Building and Construction, Value Analysis.

INTRODUCTION

In the inception stage of complex building projects the (future) facility owner or the project developer evaluates the feasibility of a new or renovated facility. Basically the question is if the facility to-be provides the client with an opportunity to do, or improve his business with returns (values) that outweigh the investments (costs) [1]. In order to answer this question we are developing a system that provides the user with a tool to play with. The game is: creating a new or renovated (improved) facility, while simultaneously evaluating the cash flow, i.e. the values and the costs.

Think for example about the inception of a new power plant. A possible client is only interested in a power plant when it can produce electricity below the market price. If a power plant can be conceived that is able to do that, bingo, business is probably on. If not, wait till the oil prices rise. The same is true for every other business and building. Though in most cases the evaluation of the values (return on investment, i.e. the possibility for the client to make money and, if so, how much) and the costs are much more complicated to calculate then in the power plant example. Take for example an office building. What is the value of its appearance? It sure is worth a lot of money if your office building is impressive, but to what limit would and should you go? And what do you decide if there are a several important performances that should be balanced together? Think about walking distances, comfort, construction time, durability, flexibility and so on.

This paper presents a first result of a research project that focuses on the value side of inception of complex building and construction projects that realise technical buildings like factories, hospitals, prisons and such. The focus on technical buildings comes from our Civil Engineering background. The intention of this research project is not to create a design for the client but to support the inception phase by stating the technical solutions with value and cost indicators. The client is supported by direct feedback and can evaluate and balance his requirements and his expectations.

After this inception process that establishes the feasibility of the idea, the conceptual design process can begin using the results of the inception phase as a reference. Differences in performance and costs between the real (concept) design and the inception design can be examined, providing the client with a good control mechanism and a realistic view on what's possible and what not.

THE THEORY

Basically the idea is to apply principles of Product Data Technology (PDT) in combination with Knowledge Technology (KT) in the inception stage [4]. In the inception stage a building (housing) is a simplification of a real building. Only the global dimensions of floors and spaces (area and volume) are taken into account. A building complex may be modelled as a set of scalable rectangular boxes for the buildings with floors and vertical transport. Choices for facade-types, roof-types and such can be made. Each building of a building complex, situated somewhere on the site, houses a set of functions in a certain (required) volume. Where exactly in the building each individual function will be placed, is out of scope; that is (part of) design.

Each simplified building is realised by pumping up the volumes and areas required for each of the functions, while constraining certain parameters like building height, or horizontal cross section. Knowledge Technology will be used to supply the building model with all the relevant systems, like interior systems, HVAC systems, etc. The knowledge that determines the values is expressed in rules or simplified input models for specialised analysis tools that are continuously running when the user plays with the system. Increasing a function (like the required number of beds in a hospital case) pumps up the required volume and the building rises, or, if the user limits the height, the building stretches. This will cause the system to re-determine the parameters that describe the required subsystems. If for example the building becomes high and slender it is quite possible that its heat loss performance deteriorates, or a totally different type of ventilation system is required. Also the foundation costs may rise, especially if a complicated pile foundation is required. Using this new information, immediately both the costs and value are recalculated. Each figure is expressed in money.

To include at least some aspects of shape the system introduces the notion of 'compact factor'. A compact factor is a measure for the layout of the total facility. If all the spaces together resemble the shape of a vertical cylinder, the compact factor can be determined by dividing the floor area of the facade with the total space. If the facility is realised as a rectangular building, a cross or star-like set-up, or as a set of distributed buildings, the compact factor increases. Each type of set-up has a more or less unique compact factor.

Conceptually it is as if we created some new kind of virtual building material that comes in the form of a Unit of Building that includes most relevant building characteristic in a scalable form. A Unit of Building transforms into a building by choosing volumes and areas for functions that play a major role in the clients primary process and by choosing types of technical solutions for major parts of the building like facades, roof, etc. Not only the buildings are taken into account, also the site area with its location, parking areas, internal roads and some process related aspects are accounted for.

THE VALUES

The first level of values follows from the clients' primary process. Which aspects are relevant, how much does each aspect contribute to the primary process in terms of value for the client, differs for each type of facility and for each client.
What we will do for this first level of values is to allow the user (client, or project developer) to express his needs in terms of performance requirements and develop a tool that gives him the power to virtually destroy and create buildings, site areas, installations and shuffle around, increase or decrease his business functions. On the fly every performance will be evaluated and met and expressed in money according to rules provided by the user. Also the costs will be continuously analysed. Values and costs will be displayed.
In the next sections we will discuss this first level of values in more detail for the case of a Hospital Inception.
The second level of values is related to the group of buildings and installations itself and therefore more generally applicable [5]. The values included in this study concern aspects like:

- Safety
- Energy consumption
- Durability
- Space Usage
- Maintainability
- Flexibility
- Comfort
- Construct ability
- Walking distance
- Accessibility

In order to evaluate for example the energy consumption, the system will include a simple heat loss calculation tool that uses a typical space layout derived from the actual set of buildings. Other aspects like strength and stiffness are only provided for by rules of thumb. In the hospital case, we use a subset of these aspects, which has to be normalised first before they can be used for the evaluation. Normalising means converting the aspects performance into a value figure for the client. An energy consumption expressed in kilo Watt is not usable for the evaluation process or for the client. Converting the energy consumption to money is more effective. For aspects as walking distances the conversion is more complex. The horizontal walking distance will probably have direct effect on the efficiency of the primary process. Think for example about all the man hours, days, weeks and month that are wasted in a hospital if the nurses have to walk continuously between different hospital functions that could be put closer together. This efficiency can be converted into values for the client.

THE PRODUCT MODEL

Figure 1 shows a picture of the simplified product model used in the system. The modelling language is UML. What you see are object classes connected with four types of relations: normal unidirectional relations (with cardinalities, single roles and open arrows), bi-directional relations (with cardinalities and roles), specialisation (with closed arrows) and aggregation (with diamonds). Each class may contain attributes (above) and methods (below), only some examples are given. The reason for using UML is that we can now generate the required Java code and use the class model as our primary representation.

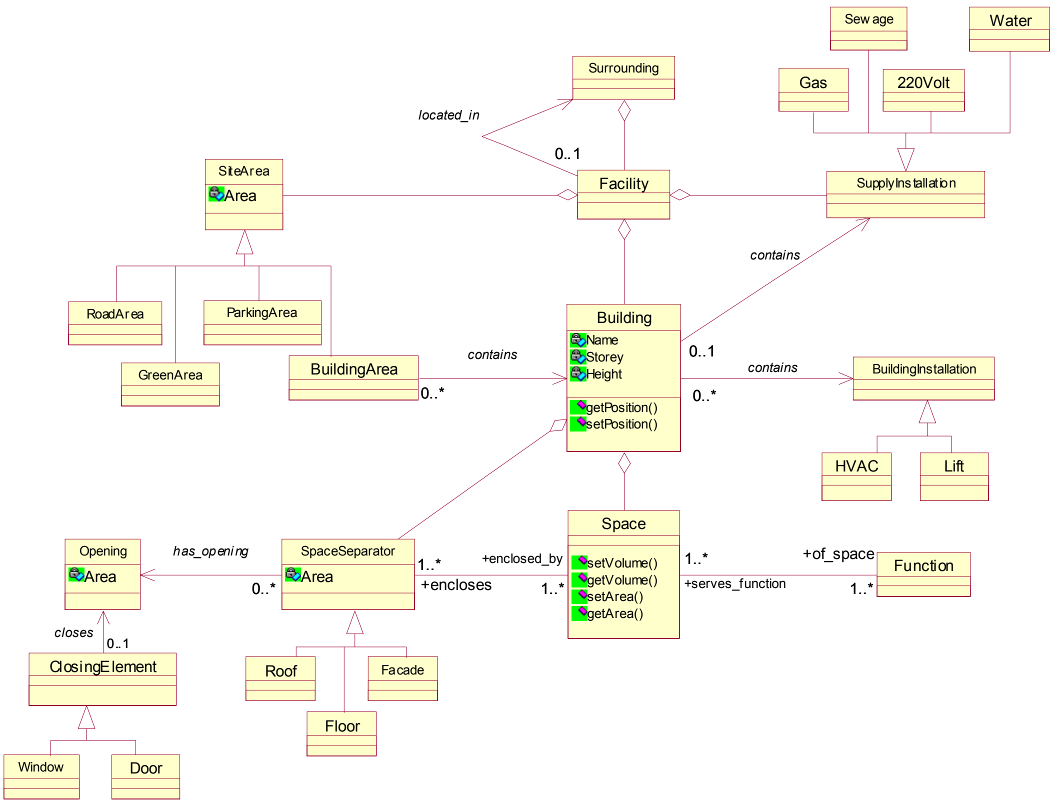

Figure 1. A simplified product model in UML. Facilities like Factories, or hospitals are regarded as collections of buildings, site areas (parking areas, etc.) and supply installations. Each building is a collection of spaces that house one or more end-user functions. Spaces are enclosed by facades, roofs and floors, and serviced by building installations like lifts, and HVAC systems.

The model says that a facility usually consists of a set of buildings, site areas for parking, green, access roads, and a set of supply installations. The buildings have building spaces enclosed by facades and roofs, and internally divided by floors. Buildings house functions (the end-user functions) that require a certain volume and floor area.
Lift systems and staircases provide vertical transport. Installations provide the required levels of light (together with the window openings), heating, ventilation, air-conditioning and such.
The product model described above is not very detailed. You will miss most objects that are for example included in the IAI-IFC model. The reason is of course that IAI-IFC is about design and this model is about inception. We also for clarity deleted most attributes and methods.

THE IMPLEMENTATION

As said, the system is written in Java and Java3D. The choice for Java is made because we want to make a nice looking GUI in Java3D. In earlier efforts we have used VRML, but then the connection between the product model and the GUI was clumsy, the user could only look at the scene and walk through it, but interaction could not be supported. We now want the user to interact with the model (use point and click and pop-up menu's) and to present him a nice looking user interface that is of the same quality as most advanced computer games of today.

In order to do this, the product model needs a shape representation and visualisation extension that maps on existing Java3D functionality. Because Java3D does only support simple topology and geometry (vertex, edge, face) and a few simple primitives like a rectangular 3D box, we had to create a shape model that includes "bounded-by" and "located-on" type of relations.

In [3] we describe in more detail how we consider the construction site and the buildings as a collection of four types of shape objects: Point-like, Line-like, Face-like and Volume-like objects, and how each object in the product model is described as a subtype of one of these shape objects.

Next we defined the required topological relations between the shape objects, as shown in figure 2.

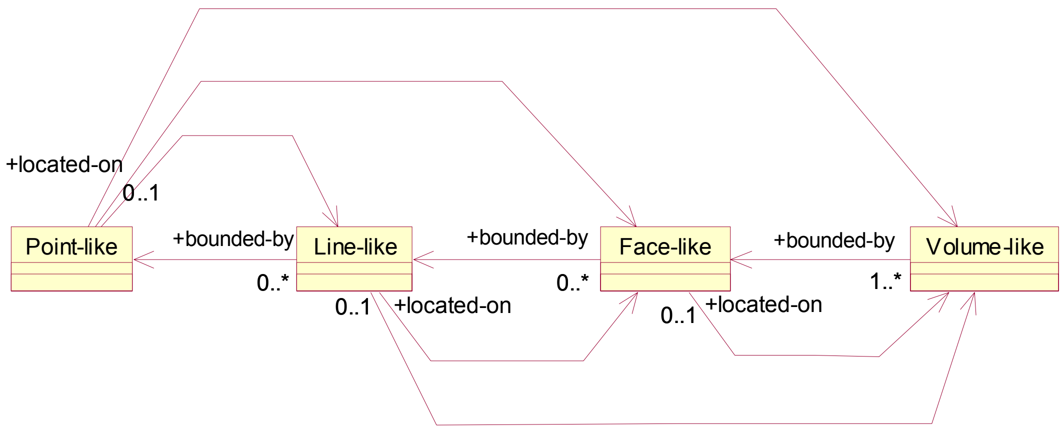

Figure 2. Shape model of the layer used between the product model and Java3D.

With the model of figure 2 it is for example possible to describe a Wall and an Opening as subtypes of Face-like object and to redefine the "has-opening" product model relation in a "located-on" relation between two Face-like objects.

Finally we created some new Java3D geometry objects that can handle the "bounded-by" and "located-on" relations of figure 2 and map them on existing Java3D functionality. One of the Java3D geometry objects we created is a swept surface element that has a polyline as its outer contour and zero, one or more polylines as inner contours (to represent openings in walls, or ponds on a site).

Figure 3 shows one of the basic objects developed. The element is made up of a general outer loop and zero, or more general inner loops, all made up of straight lines and Cartesian points.

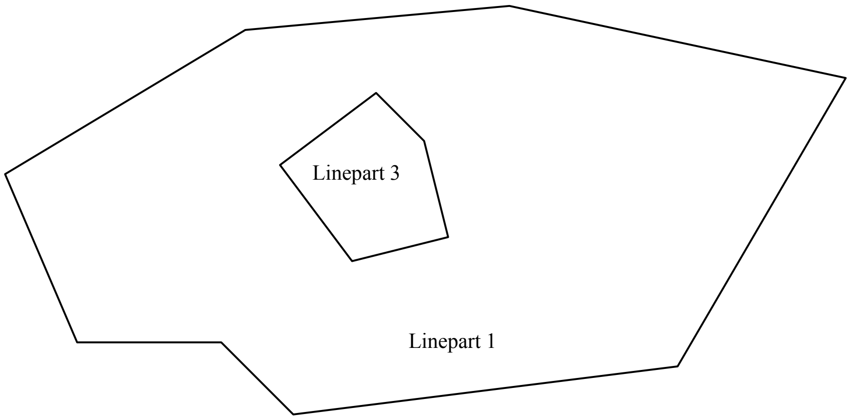

Figure 3. A shape element that maps on existing Java3D shape objects.
An outer loop with several inner loops. All simply shaped (straight, or flat).

The element can be swept using an extra thickness parameter in order to create 3D Face-like objects like solid walls. See figure 4.

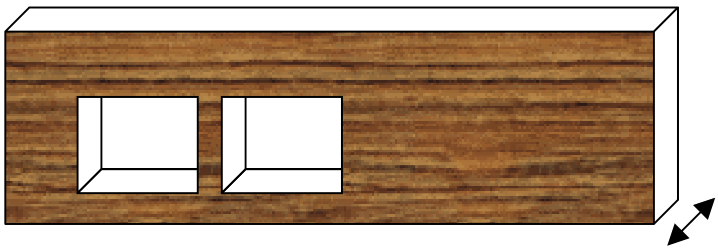

Figure 4. A 2.5D solid wall with window openings.

With the product model objects and shape model objects defined above it is possible to present a simple VR view on a building and a construction site that is very realistic and allows the user to point and click product model objects. Simple interaction by means of pop up menus is possible.

THE HOSPITAL CASE STUDY

In this case study the question was how an existing hospital facility could be transformed into a new facility that could better support the primary process (curing ill people). It is possible to compare a complete new housing accommodation on a new spot with a renovation project of the current accommodation using the Clients' criteria. Because all relevant aspects are taken into account, the value evaluation becomes a strategic basis for a design decision. The renovation project will probably cost less than constructing a new accommodation from scratch. The Client is able to check if both projects are feasible using the cost evaluation and can compare the return in investment by the value evaluation of the alternatives. The client is now able to support his decision using a value for money factor, or on the absolute netto value (value minus cost).

The interaction between the user and the system in combination with a helicopter view on the internal working of the system is displayed in figure 5. The user is able to put in the current accommodation, requirements of the wanted situation (size of hospital, etc), concept solutions (renovation or new housing facilities, etc) and an importance hierarchy regarding performances of the alternatives. This importance hierarchy consists of all neutral performances, which can be evaluated within the system, like parking space, luxury, facility, accessibility, etc. It is the client who determines which aspects he finds important for his situation. With that information it becomes possible to convert the performances on every aspect to clients' specific values.

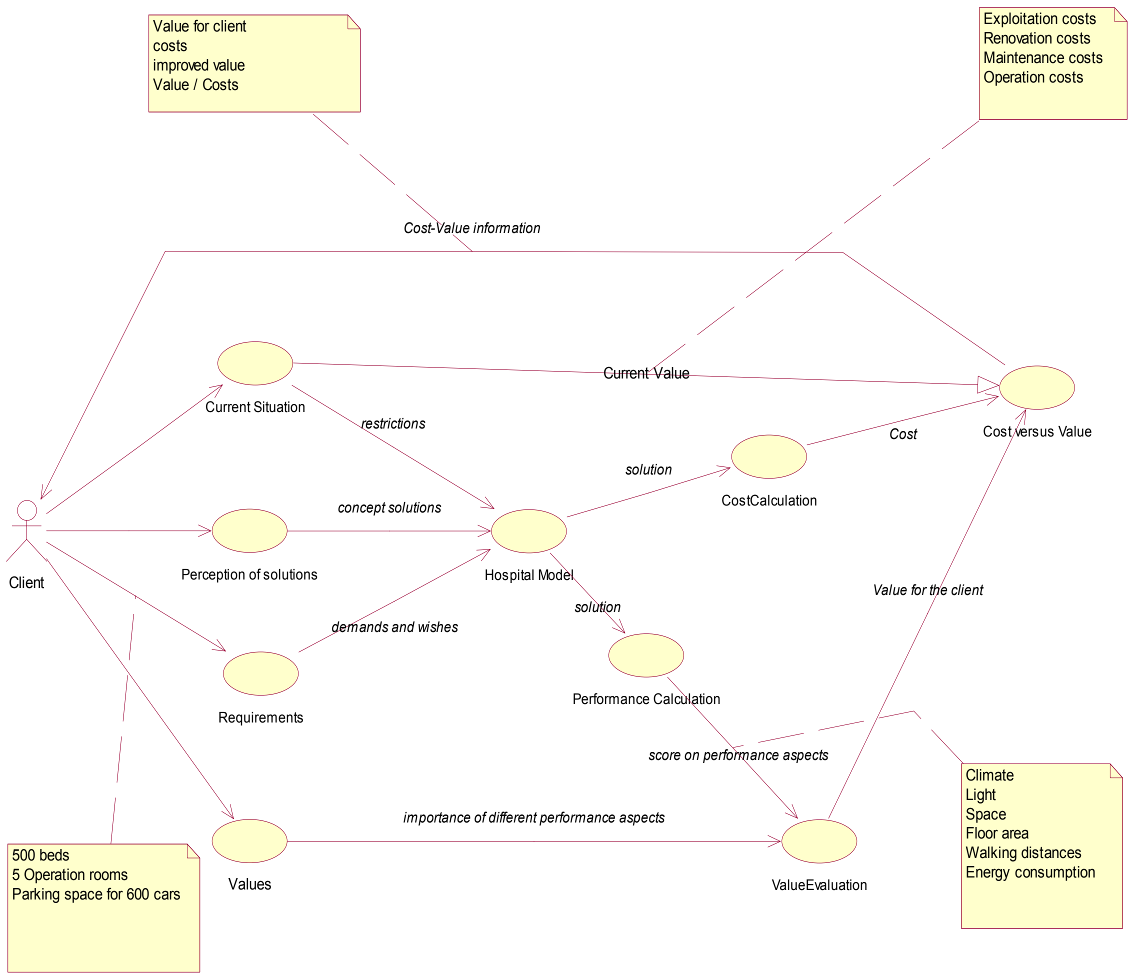

Figure 5. Helicopter view of the working of the Hospital Case Tool.

In order to have a better understanding how computer systems can support the end-user inception stage with fast Value versus Cost Evaluations, a prototype system has been developed [2]. This system strongly focuses on the end user values and is restricted to one class of facilities, namely Hospitals.

To support the point of view of an end-user, the inception tool has to contain knowledge about the end-user business process. To get a grip on an end-user business process (other then for example tendering for a power plant project) we need end-user domain knowledge. From a hospital-management point of view, the design or redesign of hospital buildings follows from changes in the primary process. Such changes may stem from changes in the population served by the hospital (more old people), new diseases (AIDS) changes in the technology (i.e. new treatments), changes in the legislation and others. In the system different objects like buildings, parking lots, green areas and such can be placed in a 3D space. Characteristics of these objects, like functions housed in the buildings can be added. (See figure 6 for the existing situation).

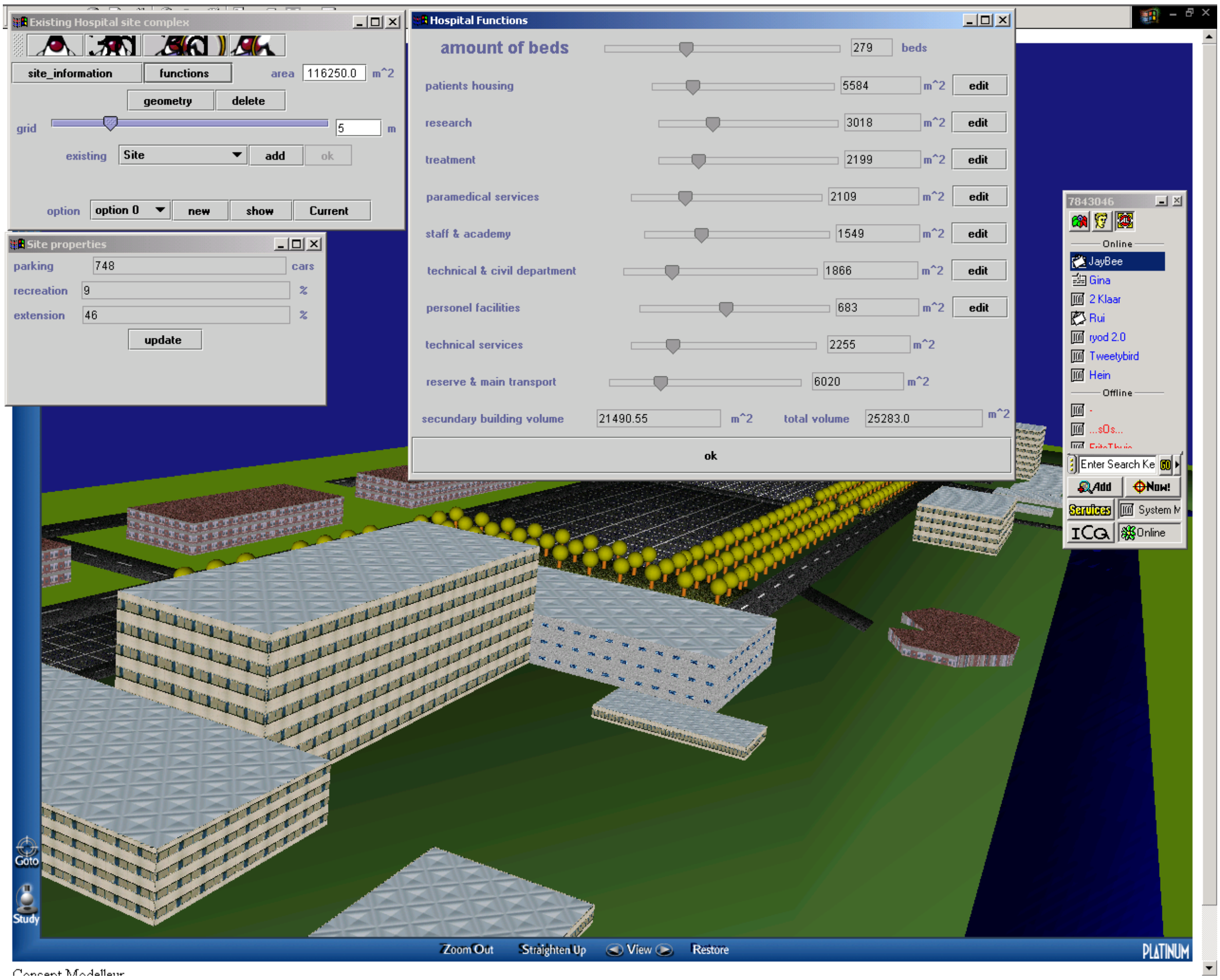

figure 6 -The existing Hospital Facility "Reinier de Graaf" in Delft, The Netherlands-

If an existing hospital is not functioning as well as it should or could, or a group of hospitals decide to service a new area, alternative strategies are developed to solve the problem. The idea is to model every alternative and evaluate its specific properties like cost, construction time, functionality, etc.

In this tool it is possible to demolish the current objects like the buildings and create new objects (buildings, parking spaces, etc). Different concepts can be compared with each other using its properties. See figure 7 for an alternative technical solution where a building is created with a compact factor on the place of the existing laboratory building.

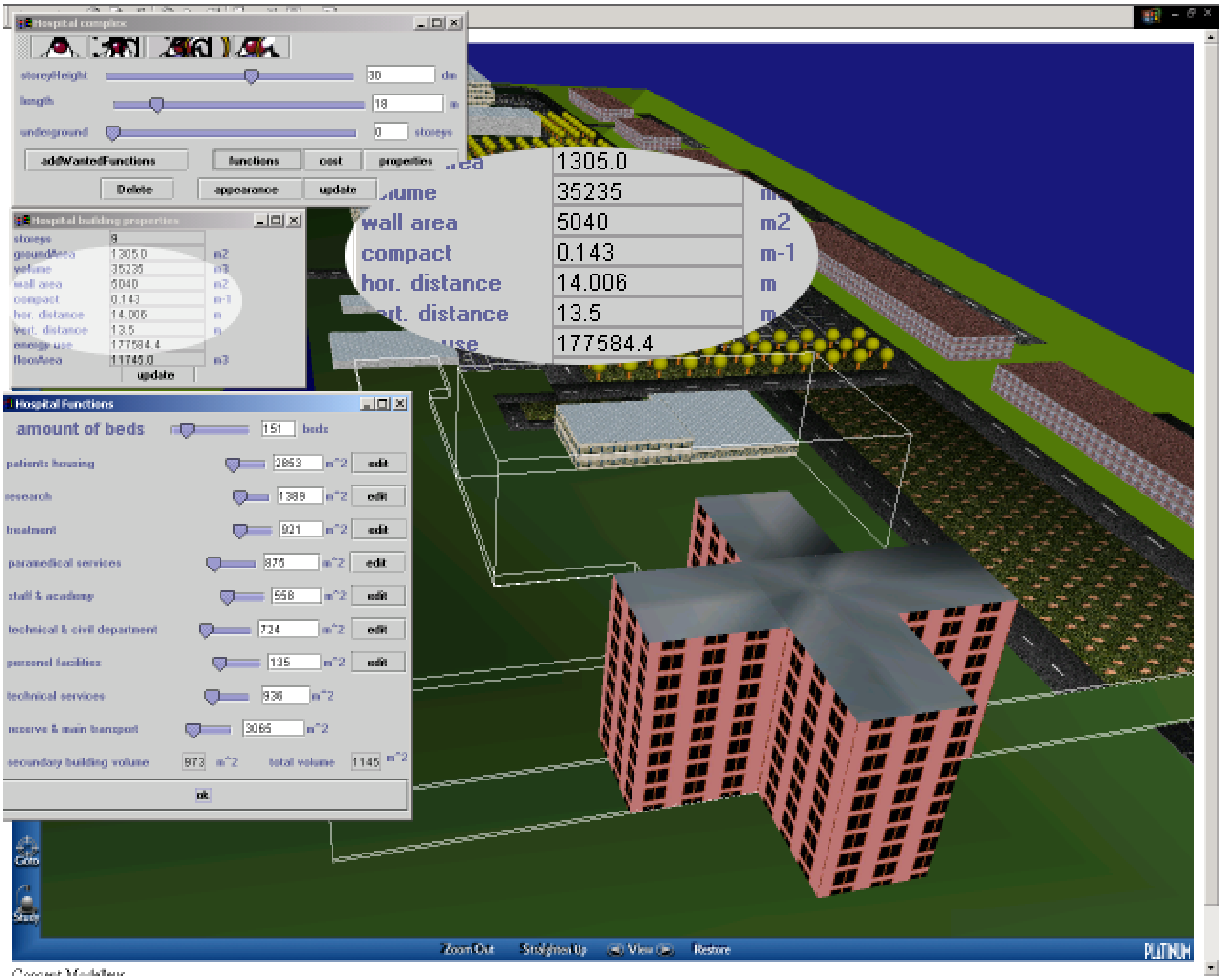

Figure 7. Some of the existing buildings have been demolished giving space to a new hospital building. A compact factor of 0.143 is used.

Using parameters like length of the building and the amount of storeys, etc. the user can manipulate the compact factor of the building. This factor is very important for aspects like daylight, energy usage, cleaning cost for windows, etc.

In order to create a better mechanism for comparing the alternative solutions, all properties can be normalized and displayed in one bar. For the normalization process, the Client has to rate the properties of the solutions. This means the Client has to specify which specific properties he finds important. Using this information the properties can be converted into values in terms of money. Now the Client is able to see the total value of the design and can compare this value in combination with the cost with other alternatives.

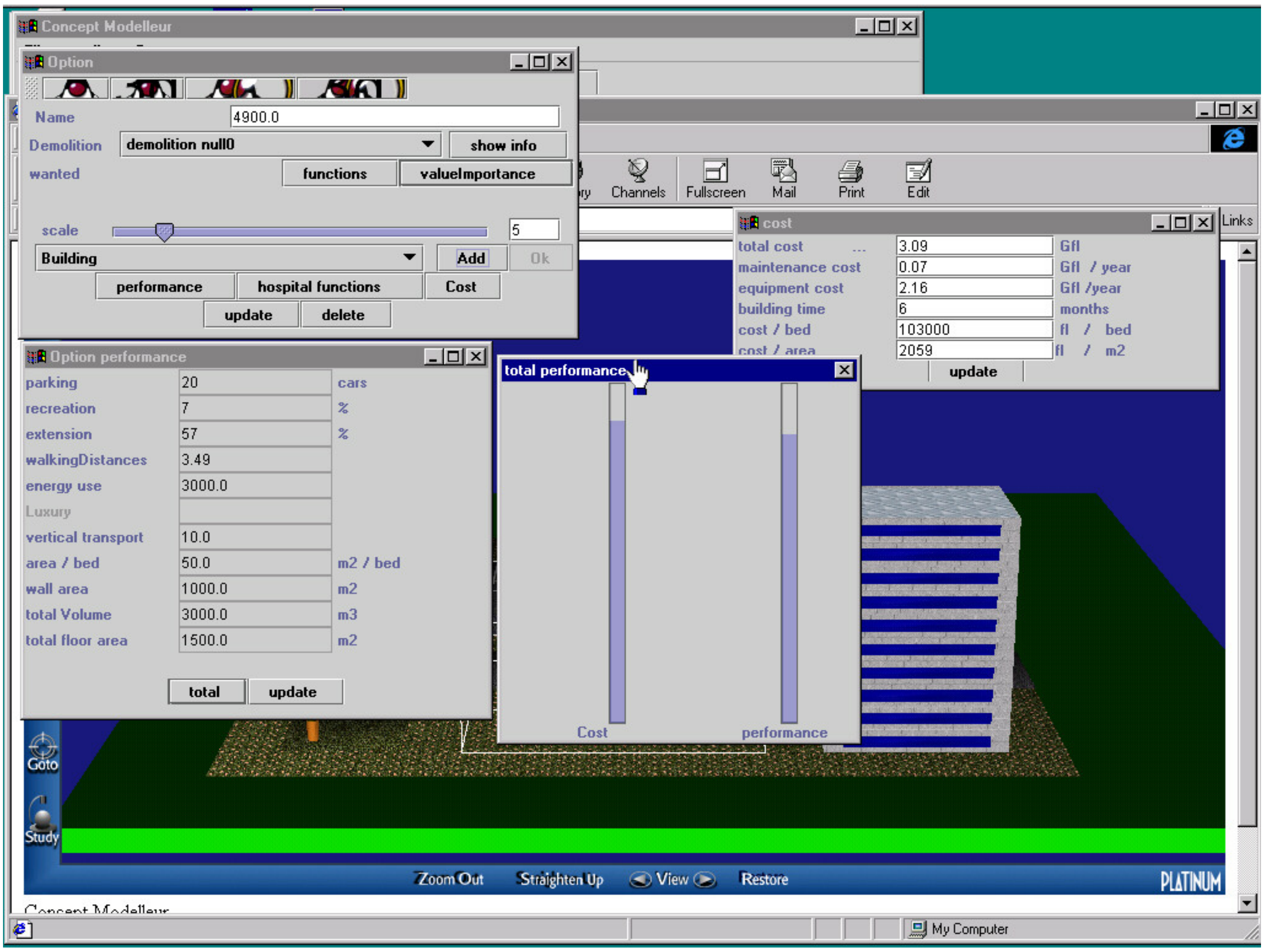

figure 8 -Value versus Cost evaluation-

The Hospital Inception Modeller showed us (and our Clients) how the end-user values can be made visible in the earliest inception stage.

CONCLUSIONS

The paper presented some initial results of a PhD research into the value-side of inception support. The basic idea is that Product Data and Knowledge Technology and the latest developments in programming languages, Java, Java3D VR User Interfaces provide the means for a new type of software that enables on the fly integrated evaluation of a number of aspects that are important for the case. In our applications we allow the end-user (Facility Owner or Project Developer) to play a creation game. If the tool is tailored to the needs of a specific type of facility (hospital, prison, supermarket, etc.) the Inception Modeller supports real-time value and cost analyses with respect to (1) the primary process of the Client, and (2) general building performances and (3) alternative realisation processes.

Of these three only the first has been implemented and tested on a case, i.e. the inception of a new hospital facility. Currently the general building aspects are looked into. We recently started with the energy consumption performance.

Each value is calculated in depth using default values and common sense decompositions, and accumulated into a sum of money that expresses what the new or renovated facility is worth. At the same time the investments and exploitation costs are also summed up. If the values outweigh the costs the Client might feel he's in business. There is a solution to his needs that earns him more money that the realisation and exploitation costs. If there is no solution the Client can down scope his ambition, which might be a jolly good idea.

This game then results in a better understanding of the problem. Clients learn to be reasonable. Also the result leads to a much improved Client Brief and a much more focused discussion with the people that promote themselves as the builders.

It is not so that the outcome of this exercise is a design, not at all! It is *input in a design process*. It is a more elaborated and balanced description of the Clients needs. That is what he wants: a new or renovated facility with these characteristics and performances and for roughly this price.

REFERENCES

[1] Tolman F. P., Ozsariyildiz S., Schevers H. 1999, 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.

[2] Schevers, H.A.J., 1999, Conceptual Design Support for Technical Buildings, Technical University Delft, Master thesis.

[3] Tolman F. P., Dado E., Schevers H., Ozsariyildiz S (2000), Implementing Virtual Reality Model Interaction over the Internet, (submitted for publication).

[4] Tolman F. P., Ozsariyildiz S., 1999, a Product Modelling Approach to Competitive Tendering in the Building and Construction Industries, ECPPM2, Building Research Establishment.

[5] Baird, G., 1995, Building Evaluation Techniques, McGraw-Hill.