IT SUPPORT FOR THE INCEPTION AND VERY EARLY DESIGN OF BUILDING AND CIVIL ENGINEERING PROJECTS

Saban Ozsariyildiz[1]
S.Ozsariyildiz@CT.TUDelft.NL

Frits Tolman[2]
F.Tolman@CT.TUDelft.NL

ABSTRACT

Despite a general agreement about the importance of the first few project stages (various sources estimate that between 60% to 80% of the total cost of the project costs are determined during the first project stages), inception and very early design of building and civil engineering projects is not yet adequately supported by information technology (IT) and knowledge technology (KT).

This paper focuses on the problems that are causing the lack of IT support in the first stages, and reports on a possible solution based on the application of Product Data Technology (PDT). The paper shows some initial experience with the development of an Inception-Modeller. The development of the Inception-Modeller takes place in co-operation with the Brite-Euram CONCUR project [1]. The system concentrates on the inception and very early design stages of technical buildings, i.e. buildings in which equipment plays a major role, such as: power plants, process plants, hospitals, and factories, etc.

Though it is still too early to draw final conclusions, it looks as if the adopted structure is ideal for inception and very early design support. It provides a means to capture, taylor and re-use construction information and knowledge in general, and of successful earlier designs in particular, thus providing a mechanism that is still missing in the building and construction industry.

Keywords: PDT, CONCUR, Inception, Early design, GARM, Knowledge representation

INTRODUCTION

In the last decades, the building industry has faced increasing demands from society. Clients ask for higher quality, shorter lead-time, lower cost, etc. Buildings become ever more complex and are situated in over-crowded, complex towns. Neighbours and companies near building sites do no longer allow disturbance of their normal way of living. And finally, the growing environmental awareness asks for better-controlled construction methods and harmless materials.

Of course, the building industries try to adapt to these growing demands, particularly by changing or adapting the way of working. They acknowledge that companies have to evolve to face up to the imposed changes. There is a growing interest for improved quality of the

---

[1] PhD. Student, TU Delft, Faculty of Civil Eng. and Geo Sciences Sec. Civil Eng. Informatics, The Netherlands
[2] Prof. In ‘IT in Construction’, TU Delft, Faculty. of Civil Eng. And TNO Building And Construction Research, The Netherlands

product and the process. Building processes are increasingly supported by integrated computer applications. Construction is becoming more industrialised by mechanisation on the site and by extensive prefabrication of building components. The collaboration among project participants is changing with the introduction of building teams, co-maker ship, design-built projects, and total quality management, performance contracts and concurrent engineering. However, these changes might not be enough to cope with the growing demands and market push.

The Construction Industry Pictured in Black and White

What we basically are witnessing at the moment is the effect of the combined arrival of information technology and new contract forms. For many decades or even centuries there has been a strict separation between client, designer/engineer, contractor and facility manager (or user). A project started by the development of a Client Brief by the client. This formal document was thrown over the wall to a designer or a team of designers/engineers that produced a complete set of formal design documents. These documents where again thrown over the wall to the contractors that produced the artefact. After the work was finished the facility was “thrown over the wall” and used and maintained by the user and facility manager.

For some decades this simple organisational picture is starting to change. The main reason is of course the fact that clients who have to pay the bill actually have very little influence over the time, cost and quality of what they buy. Not only are the facilities that they usually get more expensive then initially promised, they are also more or less in a prototype stage. The main reason is the lack of adequate information and knowledge transfer mechanisms, that support: avoiding earlier mistakes, re-use of satisfactory technical solutions, integration of different systems, re-use of available knowledge and know how.

Implementing improved information and knowledge transfer mechanisms, without IT is extremely difficult. But with the recent advent of IT in Construction, theoretically things can improve. New types of contracts, like Design-Built, lower the walls between the parties and make it possible to use for instance contractors knowledge during design (Design for Construction). The same is happening with the other walls, for example lowering exploitation and maintenance cost, is supported by Design-Built-Operate contracts and Design for Maintenance (DfM). The same holds for other walls not yet mentioned above, like the wall around the project and the environment, resulting in sustainable building and Design for the Environment.

Information technology and computer aided support systems are nowadays used by every professional, enabling a wealth of IT dedicated to: better, quicker, and cheaper execution of individual tasks, and group tasks. Product Data Technology and Knowledge Technology are, together with distributed database management systems, communication systems, and middle-ware like workflow management, the cornerstones of the powerful information and knowledge transfer mechanisms required by society.

However, one - perhaps even the most important - project life cycle stage has not yet got the attention it deserves. [2] [3] The inception stage is still largely unsupported and the advent of Incept-Design, or Incept-Design-Built-Operate contracts or their equivalents has not yet arrived. As the inception stage is the stage where all the major decisions are taken, this means that there is still a lot of room for improvements.

Inception in the Future

Though the current notion of inception has been around for many decades, it is generally being recognised that there is often a mismatch between the client's requirements and the contractor's requirements, and that a more co-operative approach should be tried. What should be done in the future is, that the client requirements are formulated *so that they lead to the most satisfactory technical solution*, i.e. the most value for the money. This goal can only be reached if the contractor is involved in the earliest possible stage.

We therefore, re-defined the purpose of the inception stage to: the definition of the client requirements in the terms of *feasible technical solutions at the lowest costs*. Feasibility should include aspects of constructability, safety and environmental pollution.

The knowledge base used in this redefined inception stage is mainly formed by the contractor's earlier experiences, extended with knowledge gained in R&D projects and incorporated in Building regulations and such. In the inception stage no detailed knowledge can be expressed, simple functions and rules of thumb will have to do.

In fact this redefinition of inception means that the contractor's knowledge becomes effective at a point of time where it still matters and where the greatest possible benefits can be gained for both client and contractor.

SOLUTION CONCEPT

In the rest of the paper we will explain the solution concept that we are currently working on. The first step is to find a structure that supports the development and usage of the multi-facetted knowledge base that covers the inception and early design of complex artefacts in a flexible way. Flexibility is mandatory, because no two countries, companies or even individuals will agree about everything that is important. Tayloring the knowledge base to individual needs should be simple.

The GARM as a Functional Model

In the GARM [4], a product (facility) can be represented as a hierarchy of so-called Product definition units (PDUs), Figure . A PDU can be a whole product, activity or resource, but also sub-system, element, component, part, or feature. For our purpose we extend the scope of the PDU to also include typical project related information about the activities and resources required. PDU now stands for Project Definition Unit. [5] The information of a PDU is given as a collection of characteristics. An Aspect determines each characteristics of a PDU. Examples of Aspects are cost, risk, quality, safety, sustainability and life cycle etc.

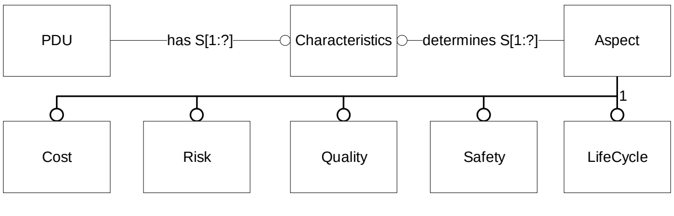


Figure 1 Definition of a PDU in Express-G [6]. A Project Definition Unit has Characteristics that are determined by one or more Aspects (Cost, sustainability, etc).

Subsequently, the GARM distinguishes two related views on a PDU: a functional view and a technical view. In the functional view the PDU is called a Functional Unit (FU) and in the technical view a Technical Solution (TS). The relations between the FUs and TSes are defined as follows:

- A TS fulfils the requirements of one or more FUs
- A TS decomposes into a set of lower order FUs

Or, expressed in EXPRESS-G Figure :

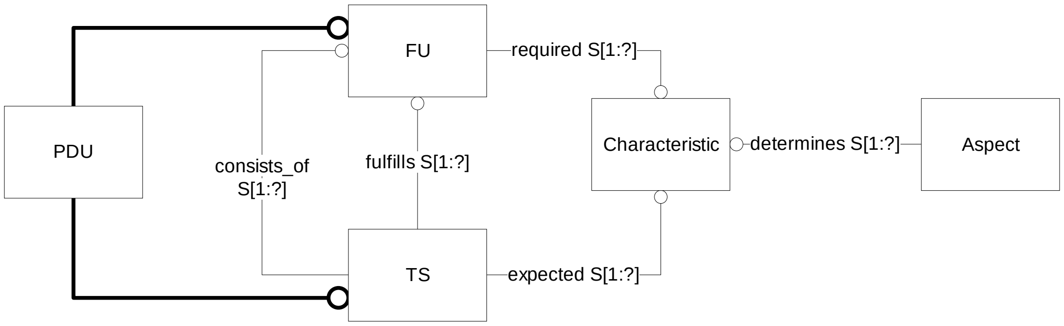


Figure 2 FUs and TSes and their relations and attributes. The model says that the required characteristics of each instance from the class FU can be satisfied by one or more instances of the class TS. The class TS contains a set of alternative TSes with expected characteristics. Note that a PDU can be either a FU, a TS, or both.

A functional unit (FU) describes the product "as required", i.e. the required functionality of the PDU (what). A technical solution (TS) is a concept that may meet the requirements formulated by a FU. It describes the product "as expected" (how). Many TSes are "standard", such as standard components, connections, features, etc, others are subject of dedicated design and engineering efforts.

Decomposition describes how a product can be divided into smaller units. Decomposition is in Express-G modelled semantically with words like: "is-part-of", "consists-of". Often decomposition limits a model to only one view on a product. However, the GARM is not limited to one decomposition, as it supports alternative TSes, FUs that are fulfilled by several TSes (a function, like stability, that is fulfilled by several components), or TSes that fulfil the requirements of several FUs (elements, like wall, that play a role in several systems).

An Example

In order to further explain the structure presented in Figure 2, we introduce a shorthand notation, sometimes called a Hamburger model. Figure 3 below shows an example of a Hamburger model of a Foundation[3].

Foundation is the required FU. Several FoundationTypes are available. The selection process can be supported by Knowledge Technology. Each chosen TS requires input values for a set of parameters. Each TS decomposes in a different set of lower order FUs. For example O-

---

[3] Only the product related objects are shown.

PileFoundation decomposes into PileGroup, GroundBeam, and FoundationSlab. For each of these FUs again sets of alternative TSes exists. Again input values are required. Again KT can support the selection process.

Note that the structure also includes the notion of interfaces. An interface says that the FU has got something to do with another FU. In this example the choice of the PileGroup might be related to the choice of GroundBeam. So, besides the fact that the structure supports the TS selection and dimensioning process, also the interaction between different FUs can be taken into account.

It should be clear that the Hamburger model as shown in Figure 3 is only useful for illustration purposes. If you think about a model of a Power Plant, or a Hospital, too many FUs, TSes and interfaces will pop up. What also should be clear however is the fact that the FU-TS decomposition provides a structure of a project that can be used as a basis for knowledge acquisition and engineering, as it supports the selection of technical solutions for given functional units, while taking account of related choices for other FUs through the notion of Interfaces.

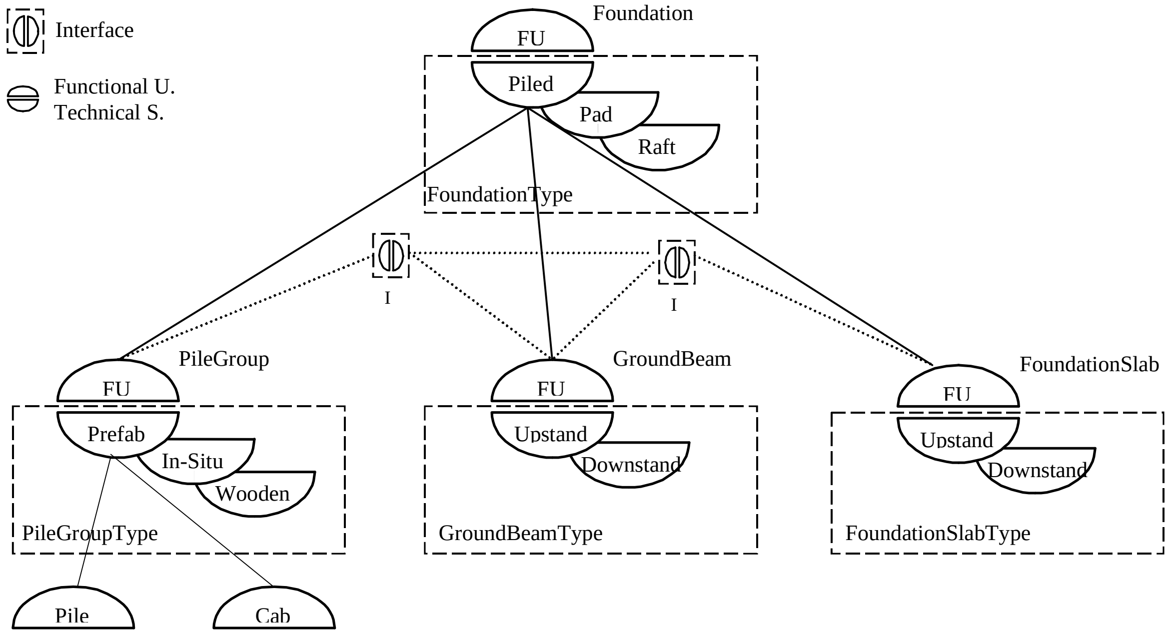


Figure 3 Part of a Hamburger model of a Foundation (FU) that can be realised by a PileFoundation, StripFoundation, PadFoundation, or a RaftFoundation, which all are instances of FoundationType. Other instances of FoundationType are also possible, like say foundations realised in earlier works. From the four types, only PileFoundation has been decomposed into a next level of objects. PileGroup refers to pile + cap. Note that also the relations between the FUs have been added, in so-called Interfaces.

Finding the optimal set of TSes for the given FUs is what inception is about. The next section will explain how the structure of the Inception Model supports the capturing, tailoring and usage of construction information and knowledge.

Knowledge Representation

A typical characteristic of inception is that there are not much crisp values and algorithmic procedures available. Most of the relevant knowledge is expressed in rules-of-thumbs and in fuzzy sets, like for instance ballpark, or square-meter cost calculations, which are formulated from cases (earlier projects). Often the knowledge is not precise, therefore allowance for vagueness and uncertainty should be made. Above all, construction projects are one of a kind projects, which means that we always will have to adapt earlier solutions to new circumstances. So flexibility and adaptability of the knowledge base are primary design considerations.

A typical expert system normally uses a fixed set of rules. In practise the fact list changes continuously, new facts are arriving and old ones being removed at all time. However, the percentage of facts that change per unit time is generally fairly small. For this reason we decided not to use the classical expert system technology, and to use a very efficient method known as the Rete (Greek for "net") algorithm. [7] The classic paper on the Rete algorithm ("*Rete: A Fast Algorithm for the Many Pattern/ Many Object Pattern Match Problem*") [4] became the basis for a whole generation of fast knowledge base engines such as: OPS5, its descendant ART, and CLIPS. [8] Therefore, in order to be flexible to modify and edit the knowledge base we choose CLIPS as our knowledge engine.

Concur

The research reported in this paper is performed in co-operation with the Brite-Euram CONCUR project. CONCUR will develop, implement and deploy integrated ICT environments for multi-partner construction projects. CONCUR focuses on electronic information sharing and exchange from the inception stage (client brief) to the tendering stage (tender documents) of technical building[5] projects.

CONCUR’s main goals are:

- to (help to) increase the competitiveness of the European construction industries,
- to develop and employ an electronic format for tendering,
- to shorten time and cost of tendering, while increasing the tender quality

During inception, cost and time are the main views on the design. In the CONCUR project, we therefore developed a FU-TS decomposition that specifically served the purpose of the tendering process. Basically, the idea is that all the relevant cost items are included in a Hamburger model, and that a knowledge tool supports the selection and dimensioning of the TSes.

Table I below, shows a part of the latest draft of the CONCUR ontology, as a hierarchy of FUs. Level 0 and level 1 covers ballpark or square-meter cost calculations and level 2, 3 and 4 covers the elements that are relevant for conceptual and bill-of-quantities cost calculations.

Note that the items in the table are all FUs for which alternative TSes can be found and that may have Interfaces with each other, and thus that the FU-TS structure described above can be applied to group all the relevant items.

---

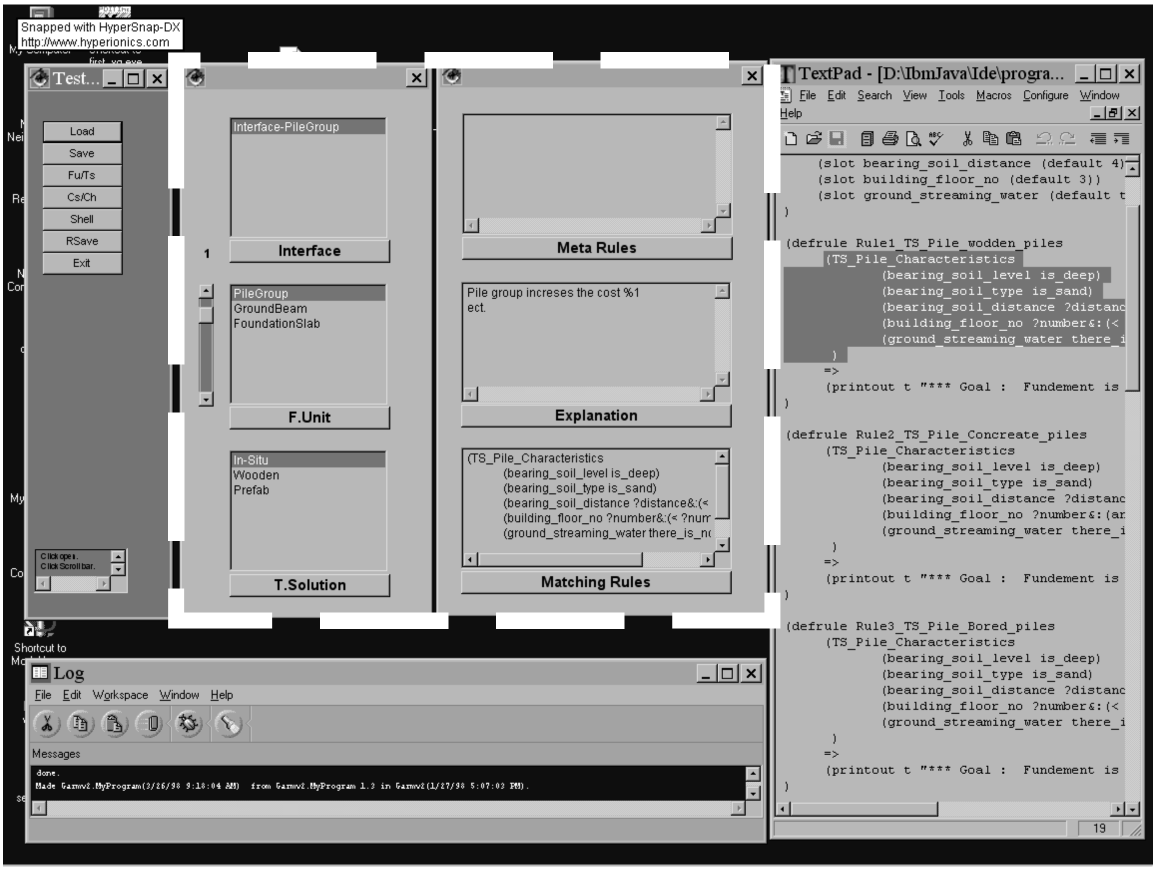
[4] Charles L. Forgy, Artificial Intelligence 19(1982), 17-37

[5] Technical Buildings include an important equipment and installation part. Examples are: Hospitals, Factories, Power Plant Buildings, Airport Terminals, etc.

Level 0	Level 1	Level 2	Level 3	Level 4
Project				
	Facility	Spaces	Activity Spaces	Vertical Access
				Horizontal Access
				Circulation Area
				Kitchen
				Office Area / Unit
				User Equipment
				Plant
				Storage
		Building	Foundation	Pile Group
				Ground Beam
				Foundation Slab
				Pile
			Super Structure	Beam
				Column
				Slab
			Façade	External Wall
				External Window Set
				Lintol
			Roof	Segment Structure
				Segment Envelope
				Segment Insulation
				Drainage
				Separation
			Partition	
			External Doors	
			Internal Doors	
			Finishes	Floors
				Walls
				Ceilings
		External Works	Roads	
			Paths	
		Services	Drainage	
			Electricity	
			Gas	
			Water	
			Telecommunications	Fiber
				Copper
			HVAC	Heating
				Cooling
			Fire Control	
		Large Equipment	Lifts	
			Excavators	
			Chillers	
			Boilers	
			Generators	
	Environmental Impacts			
	Demolition			
	Temporary Works	Temp Roods		
		De-watering		
	Finance Model	Project Cost Model		
		Cash Flow		
	Business Considerations	Operating Costs		
		Maintenance Cost		
	Contract			
	Preliminaries			

Table 1 FUs from the latest Draft of the CONCUR ontology (Technical Building 'Generator')

The current idea is to use a simple Windows based point and click user interface that allows the user to quickly enter his data and evaluate the results. The system will at any time allow the user to choose alternative TSes and parameter values anywhere in the tree. Figure 4 below shows the current user interface with the same data that is also used in Figure 3.



Pile group increses the cost %1
ect.

```
(TS_Pile_Characteristics
        (bearing_soil_level is_deep)
        (bearing_soil_type is_sand)
        (bearing_soil_distance ?distance&:(<
        (building_floor_no ?number&:(< ?num
        (ground_streaming_water there_is_nc
```

```
    (slot bearing_soil_distance (default 4)
    (slot building_floor_no (default 3))
    (slot ground_streaming_water (default t
)

(defrule Rule1_TS_Pile_wodden_piles
     (TS_Pile_Characteristics
            (bearing_soil_level is_deep)
            (bearing_soil_type is_sand)
            (bearing_soil_distance ?distanc
            (building_floor_no ?number&:(<
            (ground_streaming_water there_i
      )
     =>
     (printout t "*** Goal :  Fundement is
)

(defrule Rule2_TS_Pile_Concreate_piles
     (TS_Pile_Characteristics
            (bearing_soil_level is_deep)
            (bearing_soil_type is_sand)
            (bearing_soil_distance ?distanc
            (building_floor_no ?number&:(an
            (ground_streaming_water there_i
      )
     =>
     (printout t "*** Goal :  Fundement is
)

(defrule Rule3_TS_Pile_Bored_piles
     (TS_Pile_Characteristics
            (bearing_soil_level is_deep)
            (bearing_soil_type is_sand)
            (bearing_soil_distance ?distanc
            (building_floor_no ?number&:(<
            (ground_streaming_water there_i
      )
     =>
     (printout t "*** Goal :  Fundement is
)
```

```
done.
Made Garmv2.MyProgram(3/26/98 9:18:04 AM)  from Garmv2.MyProgram 1.3 in Garmv2(1/27/98 5:07:03 PM).
```

Figure 4 above shows a picture of the user interface of the Inception Modeller. The example is the same as in figure 3. By pointing, clicking and adding parameter values, the user provides the information from which the system generates the rules (shown on the right) that are input for the knowledge engine.

The dashed rectangle shows the ‘FU, TS, Interface’ part is shown on the left, and the related ‘meta-rules, matching-rules’ are shown in the middle. The selected FU (PileGroup) is automatically linked to top-left list where the relevant set of interfaces are listed. For a selected interface the top-right text area shows how the related meta-rules interacts with the selected interface. The list below the FU list, gives the set of alternative TSes. If one TS is selected the related matching-rules can be seen and modified interactively. The system will update the knowledge base synchronously.

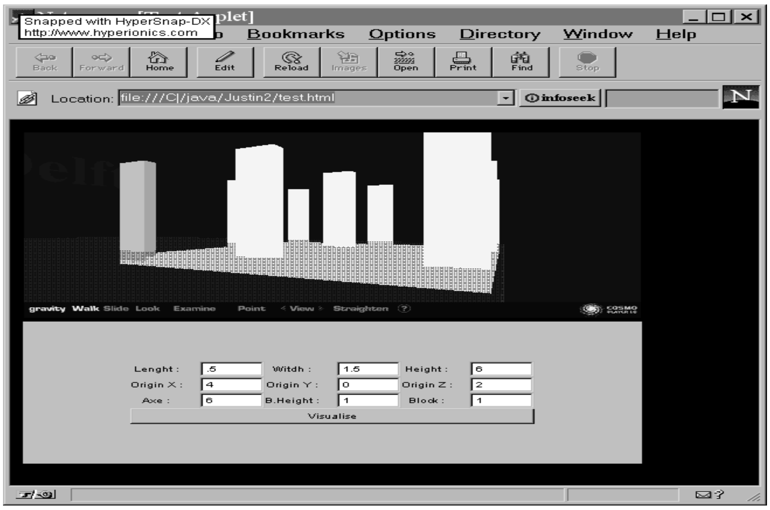
The FU-TS decomposition provides us not only with a structure suitable to capture project knowledge but also supports formalisation and re-use of cases of past experiences. For instance, the FU PileGroup has TSes such as Prefab, In-situ, or Wooden. However, a user can simply express knowledge about a previous project as: PileGroupInProjectX and by adding the characteristics of this PileGroup the system will update the case base where all past experiences are stored. The user does not need any prior knowledge more than their product knowledge. The system itself is capable of converting product knowledge to matching-rules or meta-rules.

IMPLEMENTATION

Java VRML

The system has is being developed in Java. The main reasons for using Java are: (1) Java is a nice Object based and Object Oriented programming language, (2) Java is platform independent and (3) Java supports distributed computing through internet and hyper links.

Visualisation is done with VRML, Virtual Reality Modelling Language. Figure 5 shows as an example a 3D view of a parameterised column and slab system. The user interface supports interaction. It is for instance possible that the user clicks on a column and changes the parameter values in the related window that pops up. This also means that the user quickly can evaluate different choices for TSes and parameter values.



Location: file:///C|/java/Justin2/test.html

Lenght : .5
Witdh : 1.5
Height : 6
Origin X : 4
Origin Y : 0
Origin Z : 2
Axe : 6
B.Height : 1
Block : 1

Figure 5 Example of the visualisation part of the Inception Modeller.

At the moment the tool described above is being filled with the Hamburger model that follows from Table I, and the relevant Interfaces and parameter values are added. Once this has been achieved, a real inception process can be supported.

CONCLUSIONS

Inception and very early design of building and civil engineering projects is not yet supported by adequate information and knowledge technology. The paper describes an attempt to develop an Inception Modeller that supports the inception and very early design stages, using state of the art PDT and KT. The basic idea is to structure the large amount of relevant project knowledge according to a FU-TS or Hamburger model. The FU-TS paradigm supports (1) the division of function and solution, (2) the notion of alternative technical solutions, (3) parametrics, and (4) relations between different FUs through Interfaces.

The Inception Modeller, that is being developed in co-operation with the Brite-Euram CONCUR project, supports the selection and instantiation process that takes place during the inception and very early design stages. You can see it as an intelligent check list, that has knowledge about the things one can chose or cannot chose.

The implementation uses the CLIPS system as its knowledge engine, is written in Java and uses VRML for its graphical output. CLIPS applies the Rete algorithm, which is very efficient, and supports vagueness and uncertainty in the information.

Experience with the system shows that it might well serve its purpose, but that a long way still lies ahead. Especially capturing and tayloring company construction and estimation knowledge is a difficult and time-consuming task.

Although, the knowledge engineering task might be difficult, we believe that the basic knowledge structure adopted for inception and very early design is right, thus the solution to inception support can only be achieved following the path that we are currently going.

REFERENCESES

[1] CONCUR,
Concurrent Design and Engineering in Building and Civil Engineering. Brite-EuRam R&D Project WWW Page, http://ivope.ivo.fi/concur/descript1.html

[2] Robert L. Klein,
Quality Programs and Quality Profits, Transactions from The Seventh Symposium on Quality Function Deployment, June, 1995

[3] John R. Hauser, Duncan I. Simester, and Birger Wernerfelt,
Customer Satisfaction-Based Incentive Systems Marketing Science, 1994

[4] Geilingh W.F.
General AEC Reference Model (GARM), TNO Building and Construction Research Report, 1988

[5] Luiten G.T.
Computer Aided Design for Construction in the Building Industry, PhD. Dissertation Delft University of Technology, 1994

[6] ISO (1994 b) ISO 10303-11,
Product data representation and exchange - Description methods - The EXPRESS language reference manual, Geneva International Organisation for standardisation.

[7] Charles L. Forgy,
Rete: A Fast Algorithm for the Many Pattern/ Many Object Pattern Match Problem, Artificial Intelligence, 1982

[8] CLIPS,
CLIPS Reference Manual, NASA Johnson Space Centre. WWW Page, http://www.ghg.net/clips/CLIPS.html