TOWARDS AN INFORMATION ARCHITECTURE FOR VALUE-ORIENTED BUILDING PROCESSES

Sander van Nederveen[1], Reza Beheshti[2], Edwin Dado[3] and Hennes de Ridder[4]

ABSTRACT

A significant barrier for innovative building processes is the use of fixed prices and fixed contract moments. A Value-Price-Cost model has been proposed by De Ridder and Vrijhoef (2004) for an alternative value-oriented building process in which prices and contracts are treated dynamically. A key element of this model is the use of (added) value as the key criterion instead of cost in construction contracts. Such a value oriented building process requires fundamental changes in work methods, attitudes and ICT support.

This paper focuses on the ICT support of value oriented building processes. In order to do that, the parts and relationships of the Value-Price-Cost model are analysed with respect to their ICT requirements. The main parts of the model are the value model, the specifications model, the design model and the cost model. The main relationships are the value-specifications link, the specifications-design link and the design-cost calculations link. Some of the part models and links are already commonly used in practice and also implemented in information systems. Others are less used in practice, although they have been thoroughly studied in scientific research projects. The aim of our research is to integrate the existing research results and insights and to make the resulting framework applicable for building practice.

KEY WORDS

Value-oriented building, value-price-cost, as required/as proposed, model integration, ICT for construction

INTRODUCTION

For many decades, building projects have been tendered primarily based on the lowest price. But in the last two decades, a growing awareness of the limitations and disadvantages of this approach can be observed. It was found that traditional tendering leads to minimization of profit margins, but also to minimization of opportunities for innovation. In other words, in traditionally tendered building projects, construction companies tend to stick to proven technology, do not take risks, and often try to make more profit by claiming expensive

---

[1] Assistant Professor, Faculty of Civil Engineering and Geosciences, PO Box 5048, 2600 GA Delft, the Netherlands, e-mail g.a.vannederveen@tudelft.nl

[2] Associate Professor, Faculty of Civil Engineering and Geosciences, PO Box 5048, 2600 GA Delft, the Netherlands, e-mail r.beheshti@tudelft.nl

[3] Assistant Professor, Faculty of Civil Engineering and Geosciences, PO Box 5048, 2600 GA Delft, the Netherlands, e-mail e.dado@tudelft.nl

[4] Professor, Faculty of Civil Engineering and Geosciences, PO Box 5048, 2600 GA Delft, the Netherlands, e-mail h.a.j.deridder@tudelft.nl

additional work and costly changes. Another side-effect of traditional tendering is that design changes are very difficult to manage in an acceptable way for all involved parties. As a result, design improvements are likely to be rejected because of the contractual implications. A milestone in this awareness process has been the UK Latham report (1994) in which the traditional tendering approach is identified as one of the key factors of the poor performance of the building industry in general.

The criticism towards the traditional tendering approach has led to the exploration of new approaches in which not only the cost but also the value (quality, performance) of the project is taken into account. Approaches such as performance-based building (see PEBBU), systems engineering (see US DoD, 2004) and value engineering/value management have gained attention of building professionals. An example of a model for managing building processes that goes beyond the fixed-cost approach is the so-called Value-Price-Cost model by De Ridder and Vrijhoef (2004). This is in fact a high-level conceptual model without statements regarding ICT-support. In this paper we take this model as a starting point and elaborate on information structures of the different parts of the models as well as the links between these parts. But first, the Value-Price-Cost model is briefly presented.

THE VALUE-PRICE-COST MODEL

The key point of the Value-Price-Cost model is that in any successful building project the value of the result should be larger than its price, while its price should be larger than its cost (Figure 1).

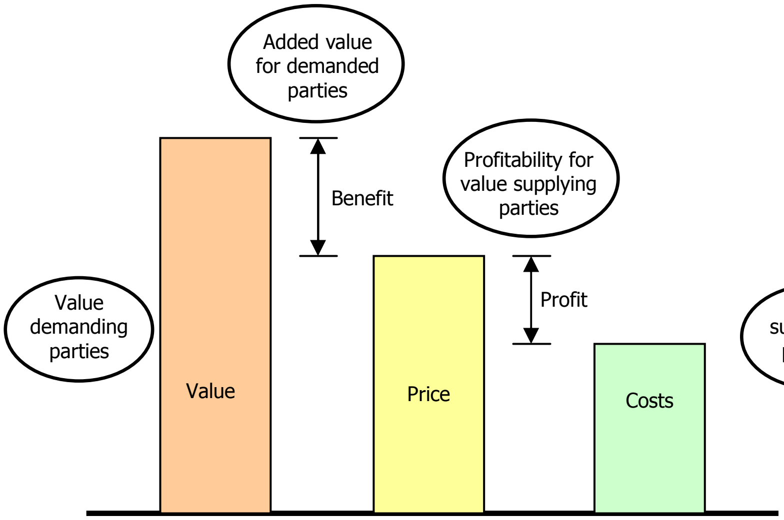

Figure 1 Basic Value-Price-Cost-model (VPC)

As Figure 1 shows, a lot of concepts can be explained using this simple model: for example benefit of the client, and profit for the contractor. The main point of the model is however, that minimization of cost should not be the goal to pursue, but maximization of added value (profit plus benefit). The model can be used to illustrate what happens when cost is reduced, when value is enlarged etc., but we will not discuss that in this paper. For explanation of such mechanisms see De Ridder and Vrijhoef (2004).

It is argued that the use of a Value-Price-Cost model is essential for improving the performance and effectiveness of the building and construction industry (Ridder, H. de et al, 2002). The last two factors in this model, namely the price and cost, have been the subject of numerous investigations while the first factor, i.e. value, remains to be explicitly defined and hence requires further investigation.

At this point, we will take a look at how design information fits in the Value-Price-Cost model. From a design viewpoint, the wanted values of the building project are first translated into requirements specifications. This translation is similar to the systems engineering notion of the translation from user requirements to systems requirements. Next, the specifications (or system requirements) are translated into a design. This translation is a very complex process (namely the design process, for which the term "translation" is actually not a very appropriate name). It is often modelled as an iterative process, in which top level requirements are translated into a top level proposed concept, which in turn is translated in "level two" requirements, etc. We will come back to this relationship between specifications and design later in the paper. Finally, the design information is used for cost calculation. The way this is done, will also be discussed later in the paper. Figure 2 visualises the position and role of the design process between "value" and "cost" as described here.

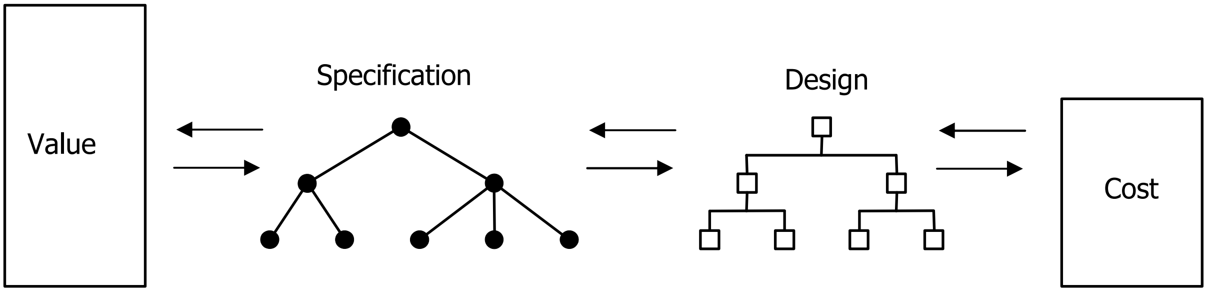

Figure 2 Design as a process to optimize the trade-off between value and cost

STRUCTURE OF THIS PAPER

In this paper, we will discuss the model parts as described above in more detail, i.e. the model parts for value-, specification-, design-, and cost information. The contents of the model parts are described, followed by a brief discussion of required and existing ICT-support. The aim is to present a more or less complete overview of the information architecture that is needed to support value-oriented building processes, and an assessment of what is available and what needs to be developed or acquired.

We will start the elaboration with design information, because there are relatively many models for design information available. From there, we will discuss the links with cost information, specification information and value information respectively, and end with some concluding remarks.

MODELLING DESIGN INFORMATION

As just said, there are quite a lot of design information models available. The simplest ones are describing only the physical structure of the artefact on instance level. In other industries, this is often called a product breakdown structure: a part-whole structure of the physical components that makes up the artefact. A building example of this approach is the so-called Object Tree approach that was used at the Dutch HSL-project (Nederveen, 2001).

The product breakdown or object tree structure has no information on component classes (such as standard parts), nor has it information on the functions or requirements of the components. However, the decomposition approach (the way the artefact is divided in parts) can well be (and is often) driven by functional considerations, leading to "functional decomposition". This introduces the question what function/functional really is. This is not a simple issue, but we will not discuss it here.

Other design information models have the notion of class (or type) and instance included. In most cases this has lead to information models on type level that are instantiated when needed. Such information models on type level contain definitions of object types such as "building", "wall", "door", "window", etc. etc. Such object types are often defined parametrically, sometimes very similar to standard parts and parametric approaches such as PLib (ISO 1995).

There are many examples of this kind of models. In the early nineties, large building product models based on ISO-STEP (1993), such as RATAS (Björk 1994) and the ATLAS (Tolman & Poyet 1995) and COMBINE (Augenbroe 1995) models gained attention. From the mid nineties onwards, the International Alliance for Interoperability (IAI) became the forum for building information models. The IAI developed so-called Industrial Foundation Classes (IFCs, see IAI (2000)), standard objects for the definition of building components and their representation in CAD-systems. Currently, the IFCs form the current standard for high level CAD data exchange and interoperability in building. In practice however, the use of IFCs is still limited, as most building companies and alliances still standardize upon a specific platform, mostly AutoCAD.

Coming back to the approach, the definition of building object types has obvious advantages as it provides the opportunity to reuse information, and prevent designers from re-inventing wheels. In fact, it fits much better to common practice where designers often reuse information from previous projects or from design handbooks etc etc. A danger of object type approaches is that people may get stuck in the type definitions. A type definition is supposed to be valid for all situations. But to achieve this, definition committees must be formed and they tend to work very slowly, as can be seen in almost any other standardisation effort.

ADDITION OF COST INFORMATION

Cost information can be modelled in an autonomous model, or as an addition of a design information model as discussed above. The simplest and most straightforward approach is to model cost as a property of any object in the design information model. In this way, there is no separate cost information model, only an extension of the design model.

In practice, this usefulness of such an approach is limited. In the best case, it can be used but leads to a significant cost engineering bias in the design information model. In other words, the model tends to be structured in a cost-driven way. Furthermore, problems arise when there is cost information which cannot be related to a specific object (for example "design cost", or "soil improvement cost"). And when cost calculations become more detailed, cost cannot be dealt with as object properties: object connections become significant cost items and become explicit elements in cost estimations, as well as activities and tasks to be carried out by on-site workers.

However, if cost information is modelled in an autonomous model, there is normally not too much difficulty in linking this model to a design information model, and there have been experiments for integrated design and cost calculation systems. Nevertheless we have the impression that the linking ("integration") of design and cost information is normally still done by hand.

ADDITION OF REQUIREMENTS INFORMATION

The link between specification and design information has been the subject of many studies. In the fifties and sixties of last century design methodology and rational design methods were popular, inspired by e.g. Alexander (1964). In the same period also systems engineering came up as an important engineering design approach. The rationality of design was based on the explicit dealing with (functional) requirements. In systems engineering literature, the terms "user requirements" and "systems requirements" were introduced along with processes to get from one to the other. While design methods lost popularity in the seventies, the interest for the explicit relationship between functional requirements and design returned as a result of the developments in information technology and design systems.

An example of a model that came out of this renewed interest is General AEC Reference Model (GARM) by Gielingh (1988). The GARM proposes a decomposition approach that includes the concepts of Function Units and Technical Solutions:

- Functional Units describe objects "as required" and have Required Characteristics.
- Technical solutions describe objects "as designed" and have Expected Characteristics.

This approach makes it possible to evaluate a design by comparing the expected characteristics of Technical Solutions with the required characteristics of the associated Functional Units.

Furthermore, the GARM proposes a decomposition tree in which:

- TSes decompose into lower level FUs,
- These FUs may have TSes,
- These TSes decompose again into lower level FUs

Etc., see Figure 3.

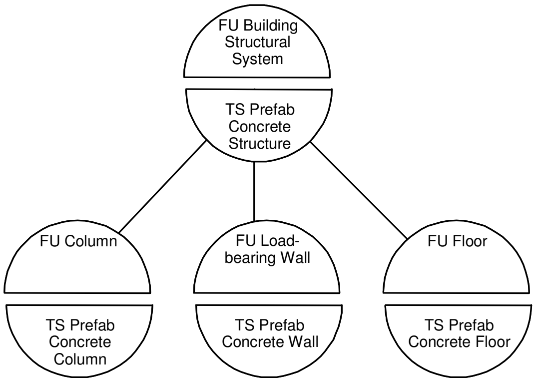

Figure 3: FU/TS-decomposition (GARM).

A similar approach, originating in the aircraft industry, but also applicable in building, is proposed by ADSE (1996). ADSE presents two hierarchical trees: a Requirements Tree and a Solution Tree. In the Requirements Tree the project requirements are defined: at the top the highest-level requirements are found, which decompose into lower level requirements and so on. In the Solutions Tree the proposed solutions for the requirements on the different levels are defined.

This means that the design process primarily jumps between the trees as follows: from top level requirements to top-level solution (the most global description of the design), from top level solution to level 2 requirements, and so on.

This is shown in a simplified way in Figure 2.2.

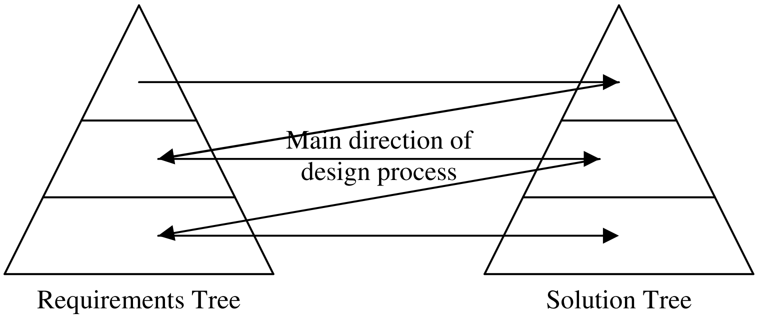

Fig 4. The Requirements Tree and the Solution Tree (ADSE 1996)

The main direction of the design process as shown by the arrows is of course not the only direction. Several feedback processes occur also, such as verification and validation processes. These processes can easily be added to the figure, but they are left out here for clarity.

As said above, this approach is similar to the GARM-approach discussed earlier. The representation with two triangles is however more suitable for this paper, since it “fits” in the overall picture of Figure 2.

From the many other studies that address the issue of modelling requirements and design information, we will mention two very interesting ones:

- Gero et al (1992) have proposed the so-called function-structure-behaviour model. In this model, the “function” part more or less corresponds to the requirements sections described above, while the “structure” (or “form”) part corresponds to the design or solution sections. The most interesting aspect of the function-structure-behaviour model is the behaviour part, which describes how the design object behaves in a certain environment (e.g. whether a structure collapses under certain loads).
- Vermaas et al (2005) approach the issue from a generic, theoretical and philosophical viewpoint and have elaborated formal definitions of concepts such as function, as well as on different approaches in working with requirements and solutions. Also a comparison is made between functional decomposition as used by engineers and the concept of decomposition in philosophy. This is done in a project called “The Dual Nature of Technical Artefacts”.

All in all, there does not seem to be a shortage of theories and models dealing with specification and design information. But looking at building practice, several shortcomings can be observed. Implemented models of requirement specification trees do exist; for example the Dutch railway organisation ProRail has done a lot in this area. Also requirement management tools already exist for quite a while, for example the RDD-100 system. Implemented models of physical object trees do exist as well (Nederveen and Tolman 2001), as do supporting tools – mostly PDM-systems or PLM-systems. But the linkage of specification trees and object trees is still done by hand.

One of our research objectives is to develop a model that links specification trees and object trees. Based on such a model it should be possible to develop a new tool that can replace or at least support the current practice of "requirements management by hand".

The authors realize that this is a hard task: the mere fact that such a link is not implemented as yet already indicates this. Part of the problem is of conceptual nature: according to our experiences it is possible to define one-to-one links between requirements and designed objects but only to a certain extent – sooner or later requirements appear that cannot be linked one-on-one to designed objects. Examples are the so-called RAMS-parameters or –ilities, as commonly used in system engineering approaches: reliability, availability, maintainability and safety. Performance assessment of these parameters is a complex task, and ICT-support for this is accordingly complex.

Anyway, in order to fulfil this hard task, it seems most appropiate to use the "two triangles" of Figure 4 as the leading model for the link between specifications and design information, since this model fits best in the overall picture of Figure 2.

ADDITION OF VALUE INFORMATION

Value information is probably the fuzziest area of the model framework we are working on. Unlike for example design information, there are not much "value information models" known to us. In the Netherlands, the large public infrastructure clients Rijkswaterstaat and especially ProRail, both related to the Ministry of Transport and Infrastructure, have done significant work in the development of checklists and models for aimed values. For example ProRail distinguishes "use value", "environmental value", "future value" etc. This work promises to be useful as a starting point for a general purpose value model for building and construction.

Even fuzzier is the link between values and specifications. The most helpful source for this area is probably the systems engineering literature. In systems engineering the translation from user requirements to system requirements is a commonly recognized step in engineering projects and there are many guidelines etc. to take this step. But it is questionable whether this literature is sufficient for the development of an information model for value information in building and construction. An important aspect of the translation of value information is value quantification. This aspect is currently investigated by Dreschler et al (2005). A research direction that could be helpful here is the area of value engineering and value management, but we did not have the opportunity yet to explore this field.

CONCLUSIONS AND FURTHER WORK

In this paper an overview is given of information models that are needed to support innovative, value oriented processes. Furthermore the relationships between these models have been discussed.

The general picture is that the different model parts are more or less common to both researchers and practitioners and ICT support for the management of this information is also common. An exception is value information, which is created in the early stages of a building project, but evaluated at different points in time later on, and eventually changed as well. For value information, a lot of modelling work still has to be done.

But even more work is needed in the relationships between the different models. While the link between design information and cost information is fairly straightforward (but still tricky), the link between specifications and design is a very challenging one and we expect some fundamental issues that must be addressed in order to improve the support of this link. Finally, the link between value information and specifications is another link that we expect to be a difficult one to model. Despite these problems (or challenges), we expect to make a step forward with our research towards adequate support of innovative value-oriented building.

REFERENCES

ADSE (1996), Handouts Workshop Systems Engineering at HSL-Zuid.

Alexander C. (1964) *Notes on the Synthesis of Form,* Harvard University Press.

Augenbroe, G.L.M. (1995) (editor) *COMBINE 2 Final report,* CEC-Joule report, Delft University, The Netherlands, 1995.

Björk, B-C (1994) The RATAS Project—Developing an Infrastructure for Computer-Integrated Construction, in Journal of Computing in Civil Engineering, Vol. 8, No. 4, October 1994, pp. 401-419.

Dreschler, M., Beheshti, R. M. & de Ridder, H.A.J. (2005) *An analysis of value determination in the Building and Construction industry.* In: Global Perspectives on Management and Economics in the AEC Sector, book 2 of 7, Proceedings CIB World Building Congress. 13-16 June 2005, Helsinki

Gero, J.S. Tham, J.W., and Lee, H.S. (1992), “Behaviour: A Link between Function and Structure in Design”, in Intelligent Computer Aided Design, Amsterdam, Elsevier, pp193-225.

Gielingh, W. (1988). General AEC Reference Model, TNO Report BI-88-150.

IAI, International Alliance for Interoperability, “Industry Foundation Classes”, Rel 2x, 2000.

INCOSE (2004) International Council on Systems Engineering (INCOSE), *“Systems Engineering Handbook”*.

ISO TC 184/SC4 (1995) *ISO 13584 Parts Library,* Geneva, Switzerland.

ISO/TC184 (1993), “Part 1: Overview and fundamental principles,” in *Industrial automation systems and integration – Product data representation and exchange*, International Standard, edition First edition 1994-12-15, ISO, Geneva, Switzerland, ISO 10303-1:1994(E).

Latham, M. (1994), Constructing the Team, HMSO, United Kingdom.

Nederveen, S van, Tolman F (2001), "Neutral Object Tree Support For Inter-Discipline Communication In Large-Scale Construction", in ITcon, Vol. 6, pg. 35-44, http://www.itcon.org/cgi-bin/works/Show?2001_3

PEBBU see www.pebbu.nl

Ridder, H. de (2006), *Living Building Atlas,* TU Delft.

Ridder, H. de and Vrijhoef, R. (2004) “Process and System Innovation in the Building and Construction Industry: Developing a Model for Integrated Value Chain and Life Cycle Management of Built Objects”, *Proceedings CIB World Building Congress,* Toronto, May.

Tolman, F, Poyet, P. (1995), “The ATLAS Models”, in Proceedings European Conference on Product and Process Modelling (ECPPM), Balkema Rotterdam, ISBN 90 5410 584 8.

US Department of Defense, Systems Engineering Fundamentals, supplementary text prepared by the Defense Acquisition University Press, Fort Belvoir, Virginia.

Vermaas, P., Houkes, W., de Ridder, J. (2005), “A Philosophical Analysis of Designing”, in: Design Research in the Netherlands, H. Achten, K. Dorst, P.J. Stappers, B. de Vries, Editors, Eindhoven, ISBN 90-6814-587-8.