CONSTRAINTS OF PRODUCT MODELLING APPROACH IN BUILDING

Constraints of product modelling

Z. TURK
Faculty of Civil and Geodetic Engineering, University of Ljubljana, Slovenia.

Durability of Building Materials and Components 8. (1999) *Edited by M.A. Lacasse and D.J. Vanier*. Institute for Research in Construction, Ottawa ON, K1A 0R6, Canada, pp. 2776-2787.

Abstract

The ultimate goal of building product modelling has been to define the data structures that could be used to describe our entire built environment. In spite of the huge size and complexity of this task, the theoretical foundations of modelling received little attention. In the paper the traditional foundations of the modelling approaches are questioned. An unorthodox perspective on the computerised representation of the real world is introduced. The view is based on Heidegger's philosophy and Winograd's critique of artificial intelligence. The author exposes the difference between the modelling of material phenomena (such as in structural mechanics) and the modelling of knowledge and information. He makes a distinction between physical models and anthropo-technical models. He claims that the latter are not objective but subjective. This is the major constraint in using product model technology in building. The author concludes that building product and process models do not model objective reality but the modeller's subjective understanding of that reality Therefore several correct but different models may exist. Based on the analysis the following information technologies look promising: simple generic models that allow room for different (even wrong) interpretations; generic software that allows uses outside the intended scope; friendly software that does not build a barrier to engineer's being in the world; non model-based software to support creative design; communication software to support commitment negotiation instead of information exchange. Complete product models and product models for creative design are not possible.

Keywords: Product models, modelling, structure, computers, expert systems, creative design

1 Introduction

In spite of extensive research there are still surprisingly few tools available, that support creative design; few expert systems and AI tools have matured to commercial levels; computable representations of standards are almost not existent; SGML based structured documents are not used; product modelling has been around for a decade yet standards are developed slowly and commercial applications are rare. On the other hand, several information technologies have received almost no attention from researchers, but are widely used in practice - for example mobile phone and fax technology, computer aided communication/co-ordination software, rather simple 2D drafting software, spreadsheets, word processing software and file based, black-box document management systems etc.

1.1 A better understanding of how humans operate is required

In this paper the author presents a philosophical background that explains this difference. It explains some of the problems that the researchers and developers in several fields of construction IT are facing. It does not solve them but suggests they could be unsolvable and that, as engineers, we should try avoiding, rather than solving them. The philosophical background that will be used is radically different to the tradition in which scientists are brought up and may, at first glance, appear unscientific: it claims that the human mind does not primarily operate rationally on mind models of the real world. Engineering software relies heavily on the models, but after reaching a certain level of bureaucratic intelligence and efficiency, the models could become an obstacle. The philosophy presented may or may not be correct, but it contributes to deeper understanding of how humans work and think, and what computers can do to help them.

1.2 Paper structure

In section "Modelling" the author defines the context of the paper - product and process modelling. In section "Tradition" the traditional approach and philosophical background of modelling are presented. Section "Hermeneutic constructivism" presents the alternative. In section "A hermeneutic perspective on product models" the author argues that product models are subjective, cause blindness, harness creativity and are of limited use. Section "Conclusions" summarises the main points and suggests what kind of product and process models could be successful.

2 Modelling

A **model** is usually defined as an abstraction or simplification of reality. A better term instead of "reality" is "original" or "universe of discourse (UoD)" because it does not imply anything real. The process during which models are created is called **modelling**. A person making a model is **modeller**. A **formal model** is a model that is formally, usually mathematically or logically defined. To act intelligently in a domain, humans do not need a model of the domain, particularly not a formal one. For example even though structural mechanics models were not existent in the Roman time, great domes have been built. Even though we still do not have a construction process model, construction processes are carried out. We can distinguish between **three kinds of models**: physical, social and anthropo-technical models.

Physical models model phenomena of the physical world - for example a cantilever, an atom etc. Models can be verified with experiments and models themselves do not change or influence the physical world: no matter what kind of theory we have on the structural mechanical properties of a beam, if we put some load on the real thing, it will deflect in the same way, regardless of our current theory. Sciences that handle this kind of phenomena are often referred to as "hard sciences".

On the other hand, **social models** are trickier. For example if a model that poor neighbourhoods have high crime rates is published, the people in the poor neighbourhood, aware of the model, might actively do something as a response to the model, that could prove the model wrong. In this case the observer and the observed influence each other, making the models much more vulnerable and the related sciences "softer".

2.1 What are building product models

Probably, because the computers in structural engineering were first used for the computations of mechanical models and because they describe physical products, we remained under the impression that when we talk about product models, these fall into the "physical" category. But at this point we must make some more distinctions.

It is generally believed that in order to be useful, computers must know about the domain in which they operate. In programs we define **digital models** of the original. Since computer programs are created to be useful for more than one original, the **schema of the model** and the model itself are separated. During systems analysis programmers try to think of every possible original the program will encounter and define a schema into which any original could fit. Instead of thinking about particular instances they think about concepts common to all instances. We can therefore distinguish between (1) **conceptual modelling** that results in a **data schema** and related code, and (2) engineering modelling, that results in data. Examples of concepts are "a tower", "a door", "a window", "quality", "satisfaction" etc.

Product and process models model products and processes such as "the Eiffel tower", "construction of the new Boston tunnel". **Conceptual building product models** define concepts about buildings and define schemata into which data about any building would fit. Data that fits the predefined building conceptual schema is called **structured data**. Building products can also be described by **unstructured data** (e.g. a GIF image). This data, since handled by a computer, fits some schema, but not a specialised building schema.

Concepts being modelled are "physical" meaning that they are defined within a "hard" physical theory. This theory, for example, defines that to calculate deflection of the beam, we must know its geometry and module of elasticity, another may require also a Poisson's number. Conceptual modelling must define a schema into which this data fits. The meaning of this data is defined within the theory.

2.2 Product models are not physical models

But **building product models** are much wider than that. They are defined as "totality of all information about a building product throughout its entire life cycle" (Bjoerk, 1989) - meaning that a union of all schemata for "hard" and "soft" theories is required. But product models are not being defined in this way - bottom up - simply because there is not a theory behind every process and behind every piece of data about a building, that we would like product-model-based software to handle. So instead of providing schemata for the data required within a self-contained, well-defined universe, we are trying to define the schemata for the concepts that are in our minds only. For example, if in process modelling we want to model a "resource" what we do is ask ourselves what we believe a resource is and what we might, during a building process, know about it. One might say that a person is-a kind of resource, one might not. There is not a hard or soft theory that would tell us who is right. Existing computer applications that the process model should support could have different notions about this issue.

2.3 Product models are anthropo-technical models

The author calls these kinds of models "**anthropo-technical**". Anthropo (and not socio-) because they are based on an individual's understanding of the UoD. Technical, because they are about the technical artefacts around us and because technology is used to handle them. These kinds of models are softest, most vulnerable and unstable. A conceptual anthropo-technical model can change one's understanding of the concepts themselves. The author calls this the **vicious circle of modelling**: "Reality" of anthropo-technical topic is a matter of convention and not of an objectively present reality: a group agrees what "really" is. For example does a "resource" include or exclude people. Models influence the mutual understanding of a topic: a model of a "resource" makes people rethink or adapt to what they consider it to be. So the model changes the universe that was observed and modelled. It should be stressed, again, that in this case the universe, has been our own understanding of the topic and not an "objective" real world.

3 Tradition

3.1 Intelligent action is based on mental representations of the real world

According to the traditional rationalistic philosophy the difference between the "reality" and our understanding of that reality is not an issue, because it claims there exists a rather simple mapping between the two. Rationalistic tradition believes that our ability to intelligently act in the world around us is due to the mental images or representations of the real world that we have in our minds. It claims that we think in terms of objects and their properties that are (more or less faithful) models of objective reality. For example, if an engineer says, "this beam is cracked", the terms "beam" and attribute "cracked" apply to some objectively present beam that objectively is cracked. Knowledge is storage of such representations. When we need to think about a problem, we retrieve a relevant representation (or, as Minsky would put it, a frame of related representations) and adapt it to the current situation.

This tradition can be traced back to Aristotle, who introduced a three-way distinction between *words*, "*experiences in the psyche*", and *things*. Early in this century Ogden and Richards (1991) proposed a similar schema in the form of the Ogden meaning triangle. It connects words, concepts and things. Each of the points on the triangle indicates a separate component that may be involved in thought or communication. The **object** is any entity from some real or imagined world about which an idea is held. The **concept** is the idea or thought of the object as held in the mind of a person. The **symbol** is an auditory, visual, or other form of utterance that is taken to stand for the object when communicated as part of a language. Any one of these components may be present without the others (Sowa, 1991). Such understanding of the human mind has provided the theoretical foundations for information and database systems development (Sowa, 1984).

3.2 Human, animal and machine intelligence are based on similar mechanisms

These ideas have also been adopted by the AI and cognitive science community (Gardner, 1987) that placed great hopes in the possibilities that computers might become just as intelligent as humans are. It has been suggested that intelligent systems (humans, animals and computers) achieve their intelligence by manipulating symbols of real world items; that symbol manipulation processes are similar in all such systems. Cognitive science postulates that because computers are symbol manipulating systems, they can achieve intelligence as well.

Programming is letting the computer know the concepts we have about the world. The symbols used in the computer (zeros and ones) will be different from what humans use to symbolise the concepts (sounds, words, graphical notation) but, so the theory claims, this is not important as long as the basic mechanisms are preserved. Using programs is mapping real world objects into computer's symbols and back. Such an approach is also the foundation for contemporary database designs and standards such as ISO-STEP(1994) and ISO-IRDS(1993). It is the foundation for the product modelling approach to computer integrated construction. Ekholm and Fridqvist (1998) have used this perspective to propose a structure of an information system for innovative design.

4 Hermeneutic constructivism

Hermeneutic constructivism (Winograd, 1995) builds on the philosophy of Heidegger (1962), findings in neurobiology (Maturana) and linguistics (Austin and Searle) and offers a radically different perspective on the workings of human mind. The workings of the mind and of model manipulation software is so different that the second really cannot achieve intelligence or creativity.

Hermeneutic means "interpretative" - meaning is not objective but achieved only through interpretation. **Constructivism** claims that truth is constructed on-the-fly and is not static. Some findings in neurobiology and psychology confirm that the conceptualisation - summarising of sensory data into higher level concepts, does not happen during store but during the recall.

The **key points of hermeneutic constructivism** can be summarised as:

- Intelligent behaviour is not based on mental representations of real world. It is pre-reflective, pre-rational, based on the being-in-the world, on "throwness" into the world, and not on being-in-the-model or being-in-a-theory.
- Problem solving is not a search in a space of potential solutions. The essence of intelligence is finding a solution outside the predefined search space.
- Communication is not information exchange - it is the negotiation of commitments among people.

4.1 Intelligent behaviour is not model based

In the core of Heidegger's philosophy is the understanding that the basis for our everyday action is the ability to act pre-reflectively when "thrown" into a situation. He claims that reflective thought about objects and properties is derived from pre-conscious experience of them as "ready-to-hand". The essence of intelligence is "**throwness**", not reflection. This is contrary to the tradition, which would consider reflective analysis of a detached observer as the basic intelligent behaviour.

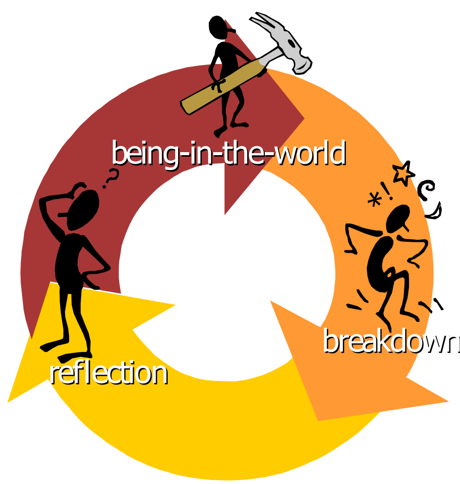

Fig. 1: The reflection loop

To hammer a nail, Heidegger argues, we do not require conscious reflective knowledge about the physical properties of a hammer and the physics of hammering. The tool is **ready-to-hand** and we just hammer the nail into the wall. Similar pre-reflective actions can also explain the so-called "intuition", "insight" or plain "common sense" that are sometimes used by the designers or engineers to explain their creative process. A good idea just pops out of nowhere, often while we managed to get out of a theory - while walking a dog or riding a bus. The solution is not a result of a systematic search in the set of potential solution. "*The essence of intelligence is to act appropriately when there is no simple pre-definition of the problem or the space of states in which to search for a solution*" (Winograd and Flores, 1997, pg. 98).

Heidegger does not claim that reflective action does not take place. On the contrary - in the case of "**breakdowns**" such as problems, difficulties we begin to examine the world in terms of objects and properties (Figure 1). We do this because, according to Frankl (1989), the main driving force of humans is the *"Wille zum Sinn"* - the will to meaning.

The models and theories developed after breakdowns have both positive and negative implications. They enable that we avoid similar breakdowns the next time - and not only we, who defined the model or the theory - others as well. They do not need to share our expertise that was required to create a model or theory, they simply need to follow the procedures as we defined.

In the Middle Ages, when structural mechanics theories were almost non existent, it took a genius to build a large. Today any structural engineer, even a stupid and lazy student should be able to design one, even an expert system could do it. Indeed, having theories and models available to insert values into, helps the routine, day to day work a lot. On the other hand models and theories cause blindness.

4.2 Models cause blindness

The very instant we begin to think reflectively about a situation and analyse it in terms of objects and properties we disconnect ourselves from the *being-in-the-world*, we are not *thrown* into a situation any more. We limit our view of the problem to the one that can be expressed by the objects and properties we have adopted, and become blind for all other possible solutions. For example in the bridge selection expert system, only the structural designs built into the expert system can be "designed". Even human engineers, when looking at the problems with a closed mind, show symptoms of blindness. Blindness is best demonstrated by parametric design where the set of potential solution is the number of permutations of different parameter values. This creates blindness for all other kinds of possible designs. The world is not made of objects with properties

Physics tells us that the world is made of atoms and he subatomic particles. Larger objects with properties are human inventions, human conventions, and do not exist outside of human procedures; procedures that did break down. A hammer does not exist as a separate object unless there is a problem with it, unless there is a breakdown. While we hammer without problems we do not experience the hammer as a special object. Because objects and properties pop into existence only if there is a breakdown, things do not have properties independent of the interpretation and the context of the breakdown.

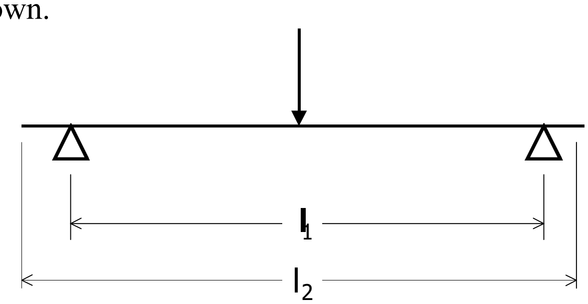

Fig. 2: Lengths of a beam

According to Heidegger, for example, a concrete beam does not have a modulus of elasticity (E). This sounds a bit drastic but when you think of it - in Roman times when they did not know about E, beams did not have that property. It became meaningful to talk about E only in the context of certain structural mechanic theories. Objectively, E does not exist; objectively beams do not have length. Length can only be established within a context where the users share a common understanding of the length of a beam (Figure 2). This does not mean, of course, that there is not a certain amount of concrete in a certain location in the universe

5 A hermeneutic perspective on product models

5.1 It explains some problems

There are several issues related to product modelling that can be explained with hermeneutic constructivism (HC):

- Most research projects define their own conceptual product models. In the light of HC it could be argued that because conceptual models are subjective, it is easy to model one's own understanding - much easier than adopting someone else's.
- Defining a common conceptual model, even within a small research team is difficult and leads to endless discussions. If conceptual models were objective and could be based on some hard scientific facts about the domain, this could be used as an argument in the discussion. But often they are not - there is no reality out there to check the models against.
- There is no way to prove a model correct or wrong, better or worse, except for its internal structure. All one can say is that it does or does not meet some requirements. This is similar to the evaluation of the works of an architect - among the proposals that meet the requirements a committee select the most "beautiful" one. In engineering, conditions of satisfaction are defined much more unambiguous - meet the requirements at lowest cost. Therefore the author claims, conceptual modelling is, to a large extent, art, and not science. If we believe Heidegger, it theoretically cannot be made into pure science.
- The conceptual product models are to an extent similar but essentially different to each other. All are said to be correct, successful, applicable, etc. Usually, in science, several people come to the same result independently.

5.2 Doubtful extrapolation of research into industry

In the 1980s expert systems were considered a very promising information technology. Today they are often used as an example for failure of a technology. What the product modelling community should be worried about is that there are some similarities between the product modelling approach and expert system techniques.

Both approaches believed that it is possible to create a computerised representation of the knowledge (expert systems) or information (product models) in a given domain. But it seems close to impossible to define complete models that would really make them useful in the practice. In research prototypes it has proven possible to define partial models or 80%-of-the-time-right expert systems. It has been argued that it is a rather simple effort to extrapolate the research prototype into an industry strength application. In the case of expert systems the critics have argued that to cover the remaining 20% the enormous amount of background and common sense knowledge would need to be computerised (Dreyfus and Dreyfus, 1988). In the case of product models the issues are similar. The modelling of data needed to support the activities for which physical models exist is unproblematic. But the modelling of data that appears in activities that are handled in an ad-hoc basis, using common sense, improvisation etc ... is not possible. Civil engineering is a domain where quite a few tasks are handled in such a way.

5.3 Product models are interpretations not representations

In section "Modelling" we have shown that product models are anthropo-technical models. They are not models of objective reality but of an individual's interpretation of that reality ... achieved in a given context. The exceptions are those parts of the product model that are defining a data model for a scientific theory.

When developing a standardised product model, a committee can achieve a common interpretation and later impose it on anyone using the standardised product model. But because the interpretation is a function of the context, domain, breakdown, culture etc. it is unlikely that the standardised model could match it all. To create complete models the entire cultural context of humanity should be modelled.

5.4 Product models should leave room for interpretation

Unambiguity and strict definition of the semantics is usually considered a positive feature of product models. But in the light of the previous section ambiguity, or rather, possibility of different interpretations, is a very important feature of product models, because it lets the human compensate for the deficiency of the model. Usually this can be achieved if the product model contains enough very generic components. For example a "line" can be interpreted as many things, including a wall. A wall, on the other can be used for few other things but a wall.

The same goes for software that implements the product models. Most successful programs are not those that try to fully model the domain in which they operate, but those that implement the most generic concepts of the domain and leave plenty of room for modification and evolution; not necessarily by adding less generic data structures but by leaving the user the freedom of interpretation of the existing ones. For example, generic tools like AutoCAD or Word, are more popular than specialised ones, like ArchiCAD or SGML editors. The seconds have a much higher level of semantics (of buildings or texts) but in.

5.5 Co-existence with unstructured data

In the early days of computing, when memory was precious, data was carefully structured. For example years were represented with the last two digits only. Structured, indexed or classified data also helped the computers to be more efficient. Today, both RAM and disk space is cheap, and the CPUs are still doubling their speed every 18 months. What remains expensive is the human effort to structure that data, both during software development and during software use. It has been estimated that in the future the amount of unstructured data handled by computers will continue to increase. We can assume that the structured part of building product data will constitute a smaller part of all data related to a building product. To be practical, product models must include or relate-to the majority of unstructured data.

5.6 Product models for creative design: a contradiction in terms

Model based software can support routine, bureaucratic processes that fit into the schema designed during software development. But can model based software support creative design? In the light of hermeneutic constructivism, the term "conceptual model for creative design" is a contradiction in terms. Only designs that fit into the model are possible. Creative ones, by definition, are those, that step out of the predefined conceptual model.

What has been regarded as an answer are "extensible" models created in such a way that the schema could be extended at runtime, such as the EDM (Eastman, 1992) and models that would support dynamic schema evolution etc. But does this really make any difference? It is not sufficient if the user can add an attribute "Poisson's number" to the schema and some rather simple constraints related to it. Of what use is it, unless there is code in the finite elements program that implements it? In other words, it takes a programmer to extend software, evolve the product model and enable a new generation of designs - routine designs within the scope of the extended software.

This does not mean that computerised tools for creative design cannot be created. If we accept the idea that that being-in-the-world is important for the human mind then we must make sure that software does not stand in the way. Therefore pen and pencil are popular with the architects and engineers in the initial stages of design - because they are not in the way, they are not limiting the creative process. Creative design can be supported by software that provides the freedom of interpretations. Software based upon complex product models is typically not like this.

6 Conclusions

A theory called hermeneutic constructivism has been presented. It exposes the limits of product modelling technology:

- Conceptual models capture a part of relevant data for the building processes. The common sense, social context, and general information should not be underestimated.

- Conceptual product models are subjective interpretations. This should be kept in mind while defining or discussing them. They are not right or wrong but appeal to more or fewer people.
- Models cause blindness. Creative work is least hindered by models that are so generic that they are not blinding or limiting.
- The success of research prototypes is relative. Product model and expert system technology faces the explosion of background knowledge and the richness of possible contexts when tried in an industrial environment.

Based on this some research directions for construction IT include:

- Coexistence of structured information and unstructured data; non-model based technologies; technologies that rely on the human to interpret the data.
- Multiple models are not necessarily bad. They confess the fundamental nature of anthropo-technical models.
- Software that does not build a barrier to engineer's being in the world but rather enhances it (like VR, telepresence, multimedia etc.).
- Communication software to support commitment negotiation instead of information exchange (Internet, groupware, collaboration, etc.).

Common to most of the listed approaches is to place the human, not product or process, into the centre of attention of construction IT. This is achieved by confessing that his view on the problem is subjective, that he operates in huge day-to-day context, that creativity can be hindered by models and theories and that the essence of human work is not processing, but functioning in a social context.

7 References

Bjoerk, B-C. (1989). Basic structure of a proposed building product model, Computer Aided Design, vol.21, number 2.

Bunge, M. (1974). Semantics I and II: Sense and Reference, Vol. 1 of Treatise on Basic Philosophy. Dordrecht: Reidel.

Dreyfus, H.L. and Dreyfus, S.E. (1988). Making a mind versus modelling the brain: Artificial intelligence back at ranchpoint, Daedalus, 177(1),185-197.

Eastman, C.M. (1992). A Data Model Analysis of Modularity and Extensibility in Building Databases, Building and Environment, Vol. 27., No.2, pp. 135-148.

Ekholm, A. and Fridqvist, S. (1998). A Dynamic Information System for Design Applied to the Construction Context, Paper for the CIB W78 conference The lifecycle of Construction IT, Stockholm, June, 1998.

Frankl, V. (1989). The will to meaning, New American Library Trade.

Gardner, H.E. (1987). The Mind's New Science, Basic Books, USA, ISBN 0-465-04635-5.

Heidegger, M. (1962). Being and time, Harper & Row.

ISO-IRDS (1993) ISO/IEC 10728, Information technology -- Information Resource Dictionary System (IRDS) Services Interface.

ISO-STEP (1994). ISO 10303, Industrial automation systems and integration -- Product data representation and exchange.

Ogden, C. K., Richards, I. A., (1991). The Meaning of Meaning, Harcourt Brace Jovanovich, New York, 1989 (First published 1923).

Sowa, J. (1991). ANSI IRDS Working Paper, Interlingua Mail, 1991.

Sowa, J.F. (1984). Conceptual Structures: Information Processing in Mind and Machine, Addison Wesley.

Winograd, T. (1995). 2. Thinking Machines: Can There Be? Are We?, Informatica, Vol. 19 No.4, November 1995, pp. 443-460.

Winograd, T. and Flores, F. (1997). Understanding Computers and Cognition, Addison Wesley (first published by Ablex Corporation in 1986).