Semantic Topology: the Management of Shape Definition

PETER H WILLEMS
Research worker

FRITS P TOLMAN
Professor

Delft University of Technology
Faculty of Civil Engineering

TNO Building and Construction Research,
Department of Computer Integrated Construction
P.O. Box 49
2600 AA Delft
The Netherlands

ABSTRACT

Over the last decade Computer Aided Design (CAD) systems have found their way in the building industry and have become a mayor tool for defining the shape of a product. However, CAD systems still do not really seem to aid the design process during all its stages. They fail to assist the early design phases, where shape definitions are not fixed yet, but exist still as roughly sketched contours.
This paper investigates shortcomings of the state of the art CAD-systems, with regard to their application in all design stages, especially within the building industry. A new approach for shape definition is introduced. This approach, called `Semantic Topology', should be able to bridge the gap between advanced product model structures and conventional geometric modelling.
This implies bringing in features for:

- adjusting the tolerancing level,
  eg, a liberal tolerancing level in the early design stage ending in the accepted manufacturing tolerances after the completion of the final design.
- defining shape constraints,
  shape constraints should clearly define the modelling freedom to define a certain shape. This feature is particular important for realising concurrent engineering.
- making geometrical structures fit for modular handling,
  modular structures, in contrast to monolithic structures, are essential to manipulate product models, standard part libraries and very large shape models.
- shape decomposition support,
  a consistent shape decomposition helps to integrate shape definition with product definition.

It is important to stress that Semantic Topology does not introduce a new kind of geometric modelling, yet it acts as an intermediary layer between a product model kernel and a geometric model.

Key Words
product modelling; shape constraints; concurrent engineering; modular modelling; shape decomposition

INTRODUCTION

Industrial automation

Several stages can be distinguished by the introduction of automation in a business environment. Firstly, mono-tasking batch-like applications are developed for enterprise functions that adequately can be defined using an algorithmic description. Then a family of matching applications is grouped into a package, which can be controlled interactively. In a further refinement, the package modules are integrated, sharing a common database.

This stage in the automation process, which will occur in various locations within an enterprise, is called island automation. The island automation stage is characterised by the increasing information flow congestion between the also increasing number of islands of automation. This information flow is either handled traditionally, *ie*, through paper based human interpretable documents, or using *ad hoc* direct translators to convert between two file data formats. In both cases, this stage in the automation process will eventually create more problems than it is supposed to solve.

The final stage in the automation process should take care of the total information infra structure, based on open standardised interfaces, common standardised product data definition schema's and open modular integrated software and hardware. ISO standards like OSI and STEP (ISO/STEP part 1, 1993) must be mentioned here, but also industry standards like UNIX, X, SQL.

Product models will contain the shared product definition data and must be stored in a neutral format with regard to all possible application that may make use of it. Neutral shape definition is one of the issues this paper will discuss.

Shape definition

What is the shape of a product and how is the shape of an object in the real world related to its computer stored counterpart: the shape representation?

First: the shape definition of a real world object is *always* an idealisation. Using mathematical concepts like straight line or flat plane a shape can be defined. This can be done in various ways, which may vary in the degree of idealisation. A shape definition in mathematical terms can be represented in a computer using, again, various techniques.

More confusing is that the shape of a product seems to change during its life cycle or that applications that deal with the shape of a product actually address different shape definitions. In a product's life cycle several stages can be distinguished:

- *as required*
  The as-required stage will apply a rather global shape definition, which may be more detailed for those parts of the product that have to interface with the operational environment.

- *as designed*
  The as-designed stage will apply a shape definition that will focus on the functions the product is supposed to fulfil.

- *as planned*
  The as-planned stage will apply a shape definition that will focus on the manner the product must be manufactured. In general, the as-planned decomposi-

tion will differ from the as-designed decomposition. Besides, the as-planned stage must define all allowed tolerance factors of each part and feature.

- *as built*
  The as-built stage will apply a shape definition of the realised product. The dimensions are measured from the materialised object.

- *as used and maintained*
  The as-used stage will apply a shape definition from the user's perspective. It will stress the outside appearance and the parts that must be inspected, maintained or replaced during the operational life time of the product.

More or less synchronous, but not identical, are the views of the different disciplines: *eg*, with regard to the building industry: commissioner, architect, structural engineer, HVAC engineer, contractor, user, maintenance firms, ...

Or the views of different applications: *eg*, visualisation, energy calculations, strength analysis, costs, bill of materials, ... (Nederveen, 1993).

Finally, a shape definition can be defined using various shape representation techniques: *eg*, constructed solid geometry (CSG), boundary representation (B-rep), spatial decomposition (octree), ... And each main technique represents a whole family of related techniques, *eg*, a B-rep can be 2-manifold, n-manifold, non-manifold, ... (Weiler, 1988).

Formal problem statement

The introduction has demonstrated that there is no such thing as <u>the</u> shape of a product. A product shape is defined by multiple shape definitions that can be represented in multiple shape representation techniques. However, all those shape definitions and shape representations refer to the same product and should be captured in one common product model. How can this be done? Or formally stated:

How can multiple shape definitions (recorded in multiple shape representations) be related to one and the same information model (product model) in a consistent and unambiguous manner?

THE MANAGEMENT OF DESIGN

Design Team

Many products, the building industry is no exception, are designed by more than one person. Each designer contributes according to his skills and discipline and the management of product development with a design team is, of course, not a new issue. Less understood is that the way the design team is managed and interacts should be reflected by the supporting information infra structure. In other words, the modular approach that characterises the design team effort must be mirrored in the way the product model is structured. Many information models show a monolithic structure, that causes tremendous problems with regard to concurrent engineering. Monolithic information models tend to reside in a single physical database and act like closed systems in the sense that they do not permit external references. Modular information models consist of multiple (sub-)information models, which may be stored in physical distinct databases. The difference with the distributive databases technique lies in the fact that here modularity is a property of the information model itself. It is not one large (monolithic) schema that is distributed over several databases (Bakkeren, 1993).

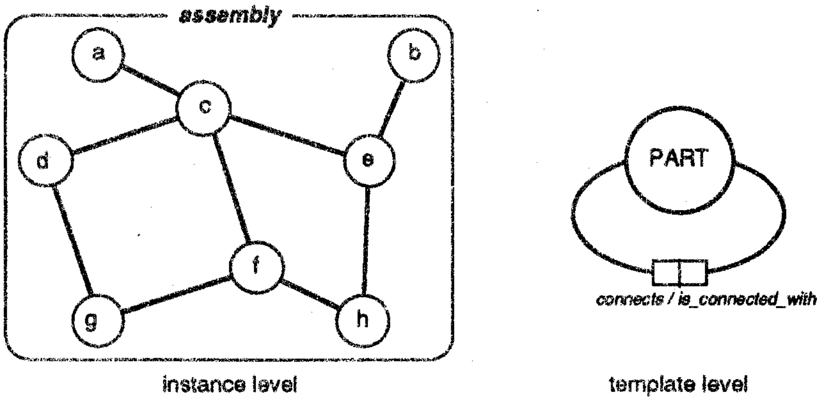

Figure 1: *Monolithic modelling. The left part shows an example of parts [circular objects a..h] directly linked to compose an assembly [blended rectangular]. The right part shows a very compact NIAM schema (Nijssen, 1989) for modelling monolithic assembly structures.*

An example of so called modular modelling is the General AEC Reference Model (GARM) (Gielingh, 1988). This generic reference model, that was part of the Tokyo draft proposal for ISO/STEP, supports modularity in two dimensions:

- *horizontal modularity*
  This type of modularity reflects the way a product, or part of a product, interacts with its environment on the same level of detail. The relations deal primarily with connectivity. By introducing explicit connection ends, internal interfaces are represented by two matching ends, while single ends represent possible external interfaces.

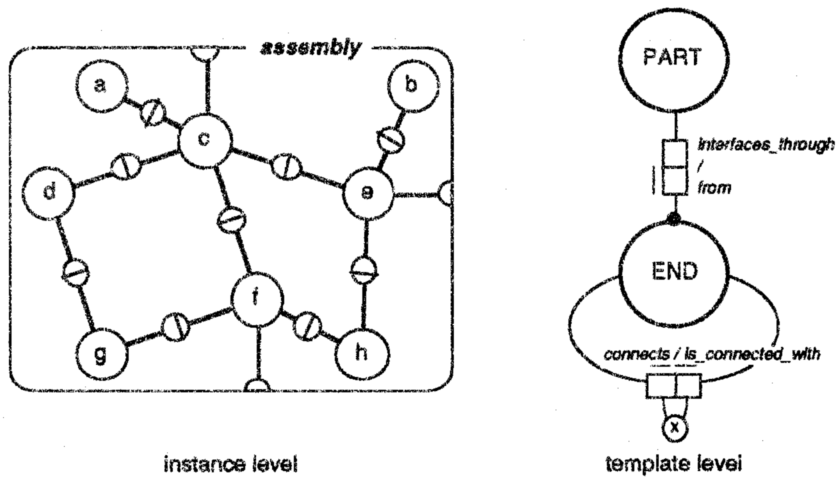

Figure 2: *Modular modelling. The left part shows an example of parts [circular objects a..h] indirectly linked to compose an assembly [blended rectangle]. The right part shows a compact NIAM schema for modelling modular assembly structures.*

- *vertical modularity*
  This type of modularity reflects the way a product, or part of a product, can be broken down into smaller or less complex parts. The relations deal with decomposition. However, the semantics of modular decomposition can be defined in various ways. A simple interpretation could be the interaction between part and assembly (a part in an assembly can be an assembly itself, etceteras), or reflect the relation between commissioner and contractor or between client and supplier. Of course, the role of commissioner and contractor could be played by different departments of the same enterprise.

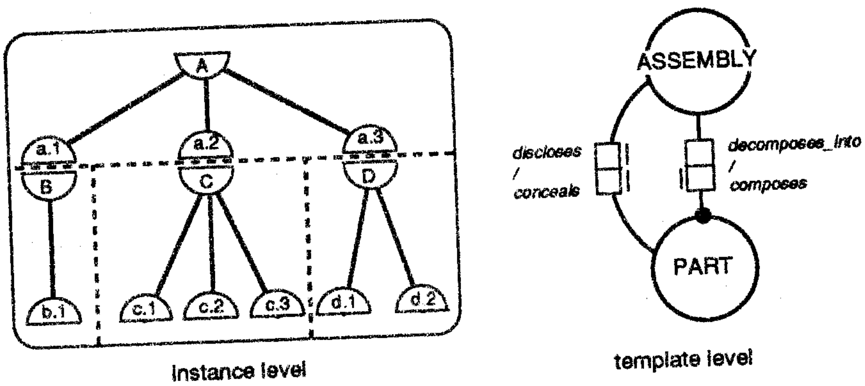

Figure 3: *Hamburger diagram: an example of modular decomposition. Here the two roles: an assembly consists of parts (1) and an assembly itself is again a part of a more complex assembly (2), are clearly distinguished. Parts are symbolised by semi circles with a flat bottom side and assemblies by semi circles with a flat top side.*

Design Scope

Using a modular structure, as described in the previous section, the design scope for each participant can be clearly defined. Figure 4 shows the various interfaces with the adjacent participants of the design team. From above the designer receives the higher level requirements for his part of the job. The result of his contribution is returned as the designed specifications. Part of the higher level requirements will be the required connections with other parts developed by other designers. This will result in designed connections (for that detail level).

In general, a particular design task is limited to a certain level of detail. Below that level is the domain of another designer. This design task could be assigned to another department of the same enterprise, or to a sub-contractor, or it is simply a product that can be bought from a manufacturer. In the last case the designed specifications are already available and can be obtained from the product catalogue.

Shape design is often an important part of a design task, applied to this structure leads to the subsequent observations:

- The higher level commissioner will indicate the important shape constraints, especially with regard to shape interfaces on the same level of detail.
- Switching roles, the designer will specify in a similar manner shape constraints for his sub-contractors.
- Precise exchange of shape data with the adjacent designers is important to fulfil the shape constraints with regard to the connections.
- The first three sets of constraints will span the modelling solution space for this particular design task.

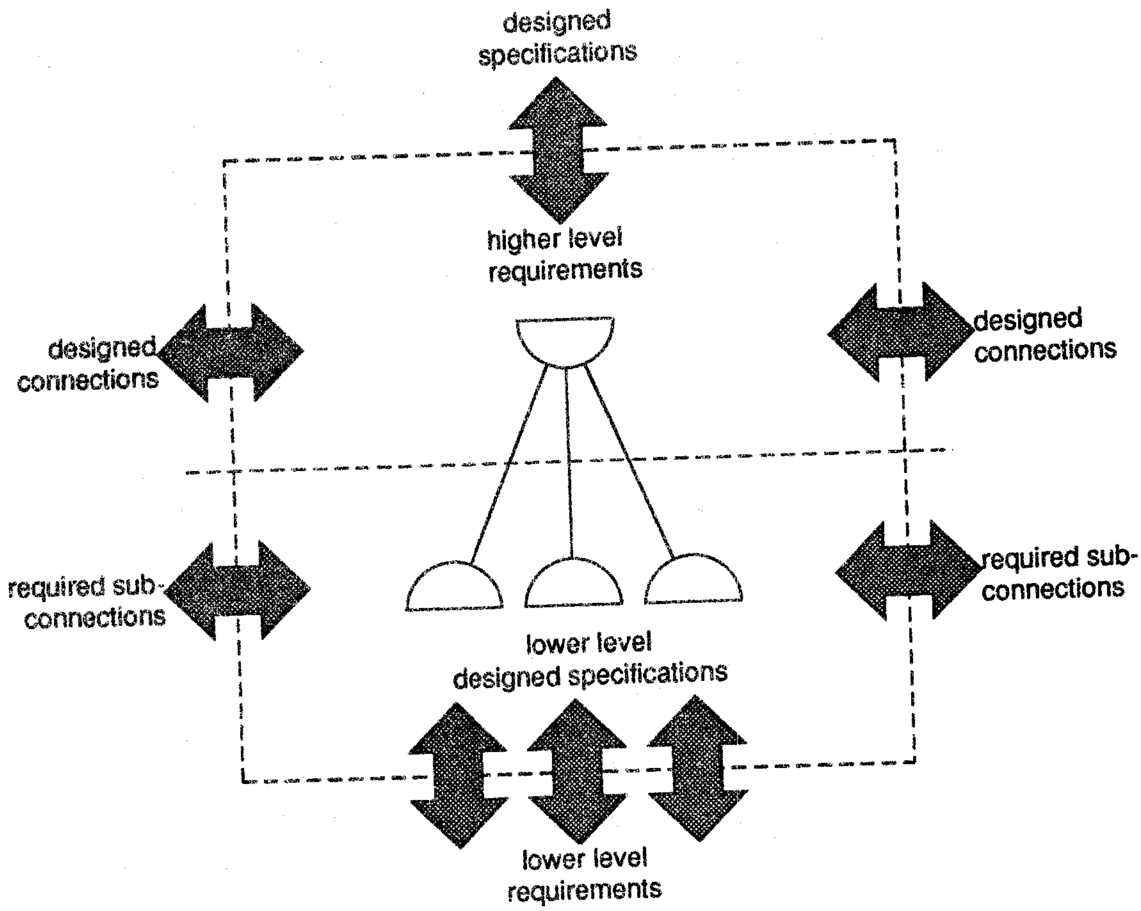

Figure 4: A single design task and its information interfaces. One higher level interface to the commissioner of the design task, multiple interfaces with adjacent designers on the same level of detail and multiple interfaces with sub-contractors.

During the execution of this process certain constraints may be strengthened or weakened, increasing or decreasing the solution space. In an extreme case the assembled shape constraints are contradictory leaving no freedom for even a single solution. Relaxing one or more constraints should create the necessary modelling space.

Present day CAD-systems are not fully capable to support the previous sketched approach. Several short comings are responsible for this situation:

- *geometry model orientation*
  most CAD-systems traditionally are designed to develop a geometric model. This is the central objective, all other information is added to this kernel structure. The introduction of this paper demonstrated that multiple shape definitions of the same product can be distinguished. The choice for a particular shape blocks all other equally legitimate shape definitions.

- *poor support for the early design stages*
  in the early design stages shape definition should be like sketching, showing that the design is not completed, yet. Most CAD-systems are very explicit and exact, they offer no options to indicate the level of uncertainty. The resulting model does not show which parts of the design are already fixed and which parts are just first attempts.

Additionally, many CAD-systems do not really encourage multiple shape editing to support this part of the design process. They are mostly applied, when the design iteration has reached a relative stable state.

- *poor support for modular modelling*
  most CAD-systems deal with only one model at a time in a single database.

- *poor support for shape decomposition*
  most CAD-systems do not support a hierarchical decomposition of a geometric model.

- *poor support for shape constraints*
  sometimes a user may specify constraints as an aid to construct the desired shape. However, shape constraints do not play a role in the resulting model.

SEMANTIC TOPOLOGY

In the previous sections an information structure has been outlined that is more adequate to support the management of the design process than current CAD-systems. Important concepts are modularity and the interfaces between the modules that represent the design scope of one designer. This section explores several issues that must be solved to control the shape definition and shape representation in this new information structure. Because topology on various levels plays an important role here (Sowa, 1984) and because representing meaning more than just a geometric model is also an important objective the research domain to bridge the gap between product modelling and geometric modelling is called here: semantic topology. Semantic topology is based on work that was published earlier under a slightly different term: meta-topology (Willems, 1988).

Shape idealisation

Shape definition is based on mathematically defined geometric and topological concepts (ISO/STEP part 42, 1993). In a three dimensional world it seems appropriate to use three dimensional concepts to represent the shape of a real world object. Indeed 3D modelling, especially solid modelling, plays an important role. However, in the early design stages solid modelling is often not suitable, because it needs much explicit data that is not available at that moment. Besides this point, there is a general need for idealised shape models for certain applications, *eg*, an energy calculation application often idealises walls and floors to two dimensional surfaces; strength analysis applications do have equivalent needs. And finally, a hierarchically structured model will need multiple shape definitions with more idealised shape models at the global levels (near the root) and less idealised shape models at the detail levels (near the leaves).

Shape consistency

By permitting multiple shape definitions there arises a need to control the mutual consistency of shape models that refer to the same object. If one shape model represents a certain object as a complex solid model, while another shape model idealises this same object to a single line segment representation, a criterion should reveal unambiguously if those two representations are consistent with each other.

Such a criterion could be defined as an assertion function, that requires two shape representations as input arguments and returns a truth value, based on a comparison of the specified shape representations. A simple criterion could be that the point set domain of one shape representation A is a sub-set of the point set domain of the other shape representation B :

$$A \supseteq B$$

Although this criterion is not very rigid, it is not quite what we want. The problem is that the super-set should define the shape constraints for the sub-set. Unfortunately, the higher level representation will be a more idealised representation and therefore turn out to be sub-set rather than super-set of the more detailed level representation, which it is supposed to constrain.

The solution must be found in the fact that an idealised representation with a reduced dimensionality, *eg*, a line segment instead of a solid, still represents a three dimensional object. For a fair comparison the idealised shape should be converted to a three dimensional shape.

Cell representation

This shape consistency issue was one of the motivations to try to adapt the current shape representation techniques to accommodate CAD-systems in supporting a consistent set of shape definitions. Starting point is the non-manifold boundary representation (Weiler, 1988), because this representation technique combines wire frame modelling, surface modelling and solid modelling in a single schema. New base element is the cell, which can be explained using its most elementary representative: the zero dimensional cell or 0-cell.

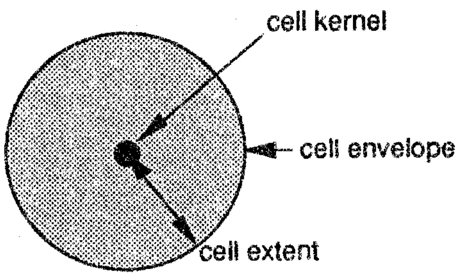

Figure 5: A zero dimensional cell (0-cell).

A cell has:

- *a cell kernel*
  The cell kernel maps directly to well-known topological concepts like: edge, face or solid. The cell kernel of a 0-cell is a vertex.
- *a cell envelope*
  The cell envelope is a closed shell that keeps at any point a constant distance to the cell kernel. The cell envelope of a 0-cell is a sphere.
- *a cell extent*
  The cell extent is the measure for the distance of a point on the cell envelope to the nearest point in the cell kernel. The cell extent of a 0-cell is the radius of the sphere that is the shape of the 0-cell's envelope.

The 0-cell is considered the basic element of the cell representation schema. Higher order cells are regarded to be compositions of 0-cells. *Eg*, a 1-dimensional cell is a chain of 0-cells. Since the location of each participating 0-cell is uncertain the 1-cell's envelope is prismatic and has a circular cross section. However, the locations of the terminating 0-cells are known, therefore the 1-cell has semi-spheres at both sides.

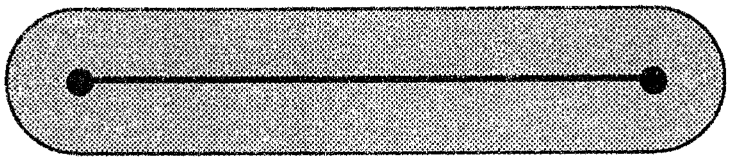

Figure 6: A one dimensional cell (1-cell) terminated by two 0-cells.

The cell representation technique adds a three dimensional envelope to a kernel shape with a dimensionality lower than three. It even adds envelopes to already three dimensional kernel shapes. In that case, *eg*, the 3-cell's envelope may refer to the allowed tolerancing level during manufacturing.

Two cells of arbitrary dimensionality can be connected in two ways: a boundary interface or an enclosure interface. A connection is typed boundary interface if:

- the intersection of the point set domain of the cell kernel of cell *A* and the point set domain of the cell envelope of cell *B* is not empty,

$$A_{kernel} \cap B_{envelope} \neq \emptyset$$

- the intersection of the point set domain of the cell kernel of cell *A* and the point set domain of the interior of cell *B* is empty.

$$A_{kernel} \cap B_{interior} = \emptyset$$

If the cell extent of cell *A* equals the cell extent of cell *B* the inverse relation holds automatically.

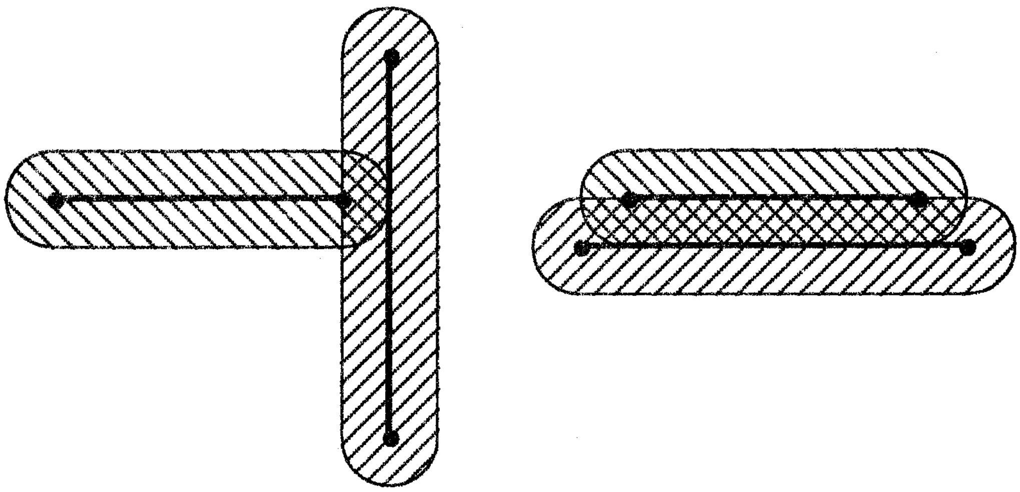

Figure 7: Two examples of (mutual) boundary interfaces.

A connection is typed enclosure interface if:

- the intersection of the point set domain of the cell kernel of cell *A* and the point set domain of the cell kernel of cell *B* is not empty,

$$A_{kernel} \cap B_{kernel} \neq \emptyset$$

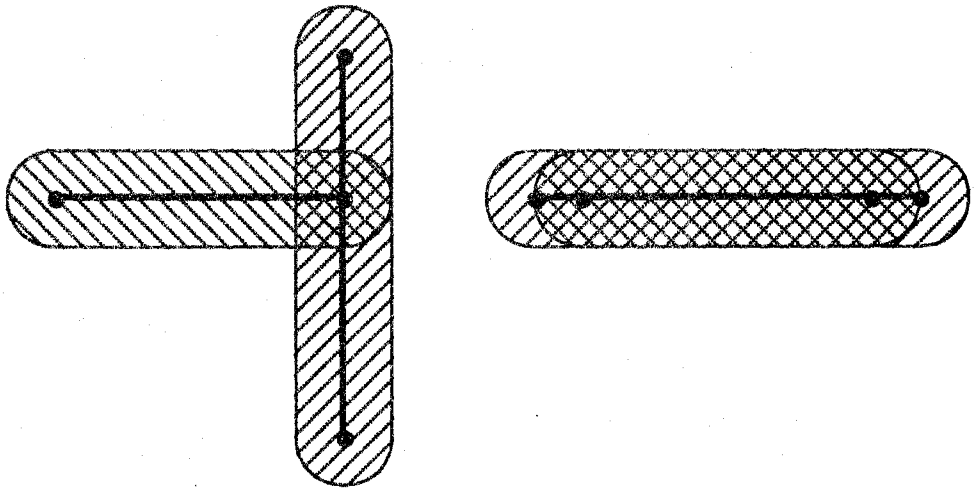

Figure 8: Two examples of enclosure interfaces.

Horizontal modularity: connectivity

A product model structure with horizontal modularity can be achieved when objects that have a connectivity relation do not refer directly to each other but indirectly by specifying ends that can be combined to an interface. Now each side of the interface can be modelled explicitly and, where necessary, in different sub-models by different designers. With regard to shape definition there is no hierarchy between the design process at either side. Both designers must agree about there common shape interfaces and the resulting shape constraints.

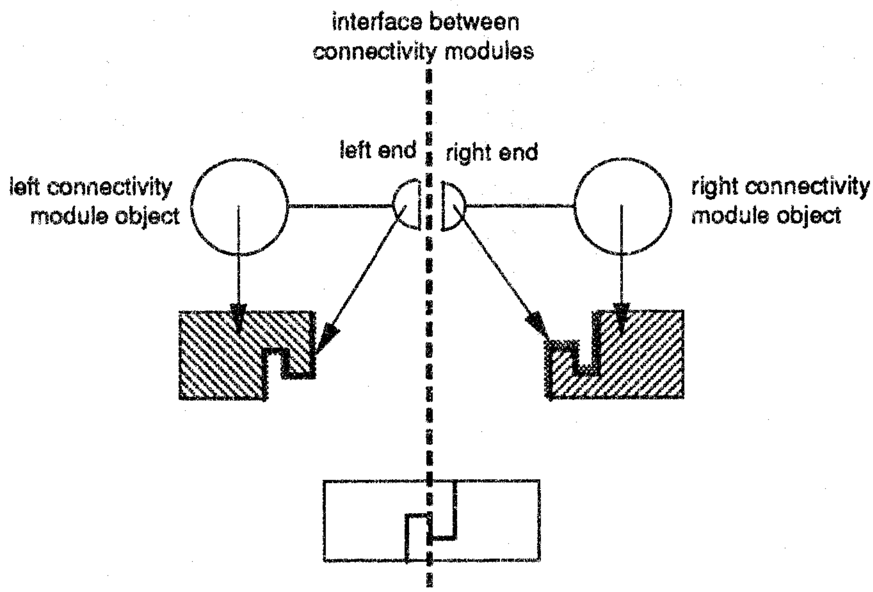

Figure 9: Horizontal modularity.

The cell representation technique can accommodate a consistent shape connectivity interface. Each end refers to a set of cells which represent the surface of the *other side*. This specifies the shape constraints that result from the connectivity interface. Interfaces can be either boundary or enclosure typed. Both ends will represent the non empty intersection point set.

Vertical modularity: decomposition

A product model structure with vertical modularity can be achieved when objects that have a decomposition relation do not refer directly to each other but indirectly through an explicit assembly entity. Then an interface can be specified between the higher level object and this assembly entity that clusters the lower level objects. Now each side of the interface can be modelled explicitly and, where necessary, in different sub-models by different designers. With regard to shape definition the shape of the higher level object constrains the shape domain of the combined shapes of the lower level objects.

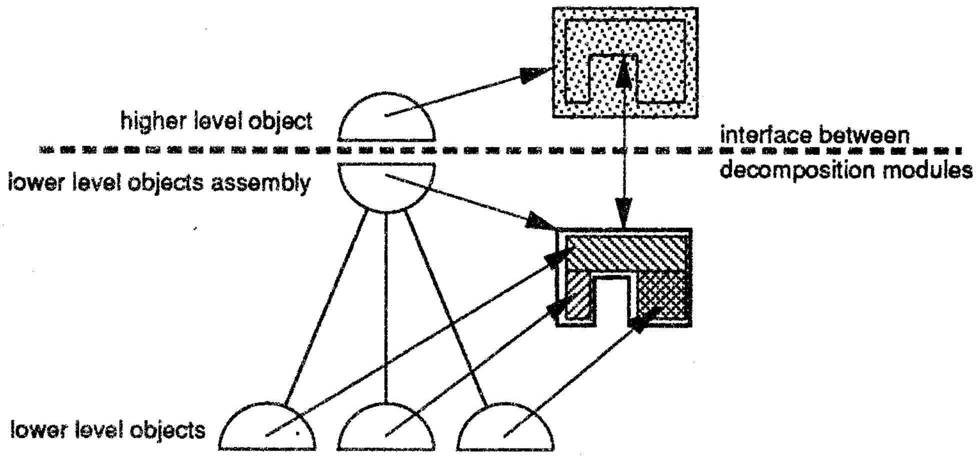

Figure 10: Vertical modularity with consistent shape decomposition.

The cell representation technique can accommodate a consistent shape decomposition. Starting from simple idealised shapes based on lower order cells with relative large extents the decomposition tree will grow to complex much less idealised shapes based on higher order cells with relative small extents. Because all cells address three dimensional shapes consistency criteria can be defined and controlled. A cell envelope of a higher decomposition level defines and restricts the modelling freedom of the lower decomposition level.

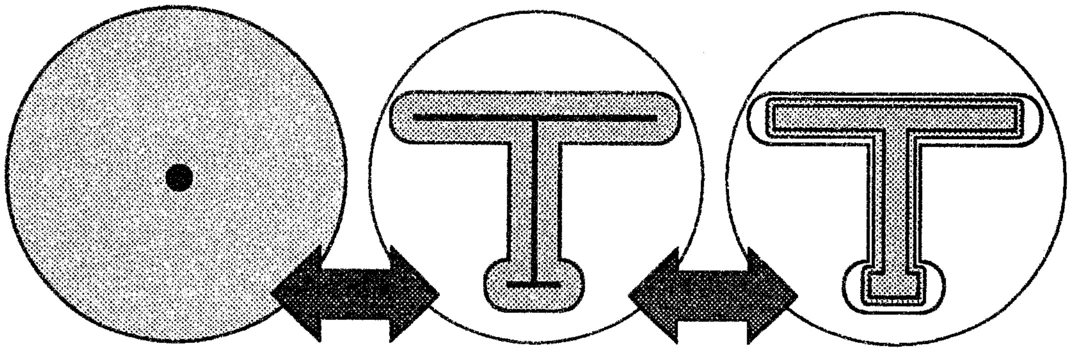

Figure 11: Levels of decomposition and idealisation, here the cross section of a beam represented as a single line segment (left), as a set of planes (middle) or as a solid object (right).

EVALUATION AND FUTURE WORK

If the shape of a product is not suitable to be used as the backbone structure of a product model another more appropriate structure should deal with this important facility that enables us to traverse the often huge collection of data. In section 2 a structure is described that not only satisfies the requirements of traversability and modularity, but also reflects the management structure of a more than trivial design project.

Banning the shape definition from the product model backbone structure has the advantage of not being forced to create geometrical entities just to store non-geometrical data. However, shape definition will always be a very important part of most product models. Over the last decades geometric modelling has achieved an impressive level addressing a shape modelling domain that is more than sufficient for most engineering disciplines. For bridging the gap between a product model structure that is not based on shape definition and geometric modelling we do not have to invent a new kind of geometric modelling. Semantic topology is a technique that introduces a number of concepts to bridge that gap that will enable us to integrate product and geometric modelling.

Future work will be intensified in the direction of a prototype implentation based on a commercially available geomentric modelling kernel. This effort will be embedded in two European technology projects (ESPRIT III). One project (ATLAS) aims at large scale engineering in the field of shipbuilding and plant design, the other project (PISA) has a fundamental scope and aims at product and process modelling.

ACKNOWLEDGEMENT

This research is sponsored by the Technology Foundation (STW) as part of the Computer Integrated Construction project (DCT99.1891).

REFERENCES

Bakkeren, Wim and Willems, Peter, (1993), *Capturing and Structuring the Meaning of Communication in the Building and Construction Industry*, draft paper CIB conference Management of Information Technology for Construction, Singapore.

Gielingh, Wim (1988), *General AEC Reference Model*. ISO TC184/SC4/WG1 doc. 3.2.2.1, TNO report BI-88-150

ISO DIS 10303 part 1 (1993), *Overview and Fundamental Principles*, ISO TC184/SC4 N181, 13 January 1993, NIST, Gaithersburg, USA.

ISO DIS 10303 part 42 (1993), *Geometric and Topological Representation*, ISO TC184/SC4 N186, 15 January 1993, NIST, Gaithersburg, USA.

Nederveen, Sander van, (1993), *View Integration in Building Design*, draft paper CIB conference Management of Information Technology for Construction, Singapore.

Nijssen, G M and Halpin, T A, (1989), *Conceptual Schema and Relational Database Sesign: A fact oriented approach*, Prentice Hall.

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

Weiler, Kevin (1987), *Two Taxonomies for Geometric Modelling Representations*, General Electric, Corporate Research and Development, Schenectady, NY 12301

Weiler, Kevin (1988), *Non-Manifold Geometric Boundary Modelling*, SIGGRAPH '87 Advanced Solid Modelling Tutorial.

Willems, Peter (1988), *A Meta-Topology for Product Modelling*. In *CIB Proceedings Conceptual Modelling of Buildings*. Lund Sweden.