A THEORY FOR THE MODELLING OF COMPLEX AND DYNAMIC SYSTEMS

SUBMITTED: June 2007
REVISED: August 2008
PUBLISHED: September 2008
EDITOR: B-C Björk

Wim Gielingh, Dr.
Technical University of Delft
email: wf@gielingh.nl

SUMMARY: *This paper presents a generic theory for the modelling of complex systems such as buildings, plants, infrastructures, organizations, knowledge bases or projects. It can be used for the description of static and dynamic systems, objects and processes. It applies to systems on any level of complexity, including simple parts, for any phase of a system lifecycle.*

Because of its generic approach, the theory supports the integrated modelling of different kinds of systems. It covers a modular definition and representation of systems and connectivity networks, supports parametric technology, and enables zero-redundant product/process specifications.

The presented theory emerged during the 1980-ies, influenced several standards for Product Data Technology, and has been implemented, used and improved for a wide variety of product-types, ranging from Buildings to Roads, Bridges, Plants, Electronic and Electro-technical Systems, Mechanical parts and Mechanical Products, Processes and Organizations. Four cases of application are discussed, and some background information is given about changes relative to earlier versions.

KEYWORDS*: system theory, product modelling, process modelling, nD modelling, Product Data Technology.*

1. INTRODUCTION

1.1. Background

The presented theory originated about 28 years ago. It was inspired by product modelling through parametric technology - in particular feature technology, such as applied for the Computer Aided Design of mechanical parts. Feature technology aims at the implicit, representation independent definition of shape and material properties of parts. Feature technology is not intended for the description of complex systems, so that it has to be extended for that purpose. A first successful application was developed and implemented in 1982 for a manufacturer of interior walls of office buildings. This application was still based on 2-dimensional non-manifold topologies (i.e. walls being idealised as 2-dimensional structures) but was soon after up scaled to 3-dimensional non-manifold topologies, where planes represent floors, roofs or walls and volumes represent spaces.

The theory was extended for the description of entire buildings, and was submitted as a contribution to ISO standard 10303 (STEP = Standard for the Exchange of Product Data) in 1986. Because of its generic nature it was adopted as the General AEC Reference Model (GARM) (Gielingh 1986, Gielingh 1988). The final architecture of ISO 10303 did not accommodate a slot for GARM, so that it disappeared as part of the final standard. Parts of the GARM reappeared in the integrated resources for STEP, but are hardly recognizable as such.

Despite its disappearance from STEP, some of the principles were cited and used in several follow-up initiatives, such as research projects, PhD studies and new initiatives for standards. A reasonably complete software implementation was done at the research organization TNO where the author worked until 1995. Certain parts are referenced by the IAI/IFC standard for CAD data exchange (IFC 2007), STABU Lexicon (STABU 2007), a Dutch national standard for Building specifications, and PeBBu, an international initiative for building performance specifications (Foliente 2005).

Through the years, the GARM concepts were tested and further improved. But they were not published under that name anymore. This may explain why many improvements remained unnoticed to the research community.

Important milestones were the European projects IMPPACT (Integrated Modelling of Products and Processes using Advanced Computer Technologies) and PISA (Platform for Information Sharing by CIME Applications). In IMPPACT, the theory was applied for a data sharing environment for discrete parts manufacturing, which included integrated design, analysis, process planning and CNC production (Bjørke and Myklebust 1992, Gielingh and Suhm 1993). In PISA, the concepts were applied for a collaborative design environment for car manufacturing (Gielingh et al 1995). Although the latter projects were not construction oriented, they enabled verification of the theory in industries that are advanced with the application of product modelling.

In a more recent study, it is shown how the concepts support a methodology for design, construction and lifecycle management, called Cognitive Engineering (Gielingh 2005) or Cognitive Product Development Gielingh 2008b).

This paper presents an updated description of the theory which is substantially more complete than the versions published as GARM in the 1980-ies, but is more comprehensible than the one published in 2005. It may be referred to as Extended GARM, or GARM-X.

1.2 Not just a Reference Model

Previous versions of this theory have been published in the form of an information model. Because of its level of abstraction, it has also been called a 'reference model'. Within the context of object-oriented thinking, top-level information and knowledge models put constraints on more specific models. Further, there is very little room 'on the top', just enough for a single model. This has caused, and is still causing, strong debates in the domain of standards development (Björk 1995).

Apart from the GARM there have been several other reference models for construction, such as the Building Systems Model (Turner 1987), RATAS (Björk 1994), IRMA (Luiten et al 1993) and the Building Construction Core Model for STEP (Wix et al 1994). Noteworthy is also the Epistle Core Model, which is part of ISO 15926, a standard for the modelling of process plants. Generic reference models have also been published to enable semantic interoperability of ontologies, such as the Suggested Upper Merged Ontology SUMO (SUMO 2007); the Basic Formal Ontology BFO (Smith, Grenon 2006), the Generalized Upper Model GUM (Bateman et al 1995) and the Descriptive Ontology for Linguistic and Cognitive Engineering DOLCE (Masolo et al 2003).

A framework for information models, which is proposed by (Björk 1995), places models on different levels of abstraction. But abstraction is a form of perception. Hence, there are many different kinds of abstraction possible, which makes it difficult to compare models that are placed on the same level. It is even more difficult - if not impossible - to adopt just one as being the only right one. These different abstractions are the main cause of interoperability problems with object oriented or ontological methods. This subject is discussed in more detail in (Gielingh 2008a). Given these limitations and fundamental conflicts between theory and modelling methods, the theory that will be presented here is not expressed in the form of an information model.

Apart from levels of abstraction, it is also possible to categorize abstract information models into (1) implicit (parametric) models, (2) explicit models and (3) taxonomies. The GARM belonged to the first group, the Turner model to the third. The IRMA and BCC reference models and most PDT standards (STEP, IAI/IFC) belong to the second group.

A feature that distinguishes GARM from other reference models, is that it does not predefine object classes, attributes (or properties), relationships or functions. Thus, it does not mention buildings, rooms, walls or piping, nor materials (concrete, brick), nor characteristics such as weight, typical mass or temperature, nor mechanical connectivity or conductivity. These can be defined on another level of abstraction, or can even be considered as data. A somewhat similar approach is followed within ISO 15926. This standard has a core model which is remotely comparable with - and partially based on - GARM. Classes, attributes and relationships are incorporated in this standard by a separate reference data library (RDL).

It is possible to implement and use the GARM, even without having more specific information models available. Several cases will be presented here where the GARM theory is implemented and applied in industrial practice without additional specifications. 'Object classes', attributes and relationships are then part of company or product specific standards and are treated as data. It is in this form that most implementations were made. Relative to object-libraries or taxonomies, the GARM-theory would be a "meta-model".

1.3 Purpose of the graphical notation

Please note that the graphical depictions of schemas in this paper are intended for explanatory reasons only. The graphical notation is not intended for direct usage as a method. This notation was developed for explanation of the decomposition and networking principles, for which existing graphical methods are not suited. Real models of products and processes become rapidly very complex in this notation; actual applications have to be supported by information processing systems.

1.4 Structure of this paper

The first part of this paper covers the theoretical aspects. Chapter 2 describes the modelling of systems, both in terms of hierarchical structure and networked structure. Chapter 3 addresses the modularization of system models. Chapter 4 discusses the organisation of knowledge about systems, including the distinction between static and dynamic views. Chapter 5 describes in more detail topological networks. Up to this point, the article aims at presenting a coherent and consistent theory. The current theory evolved however from earlier versions. Deviations from these initial versions are discussed in chapter 6. This chapter provides also a bit more background information about key principles, such as the 'Hamburger' concept. Chapter 7 addresses a few case studies and applications. Chapter 8 draws conclusions.

2. ON SYSTEMS

2.1 System composition modelling: wholes and parts

Systems theory is based on the concept of wholes that consist of parts. Each part may on its turn be a whole, consisting of parts.

Systems are often modelled and depicted graphically by means of an inverted tree structure. The top (or root) of this inverted tree depicts the whole; the branches depict the parts; see also figure 1.

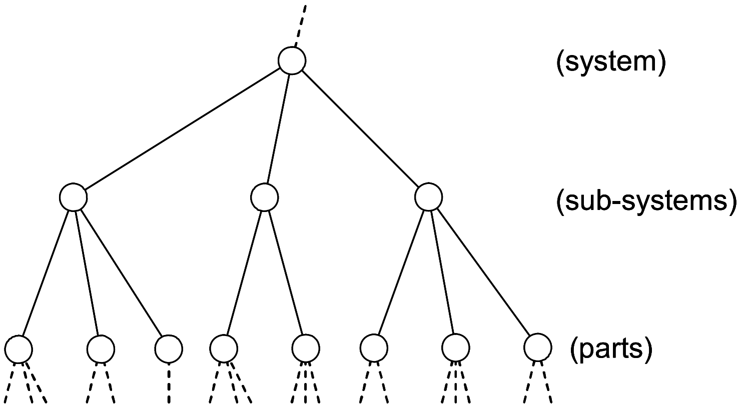


FIG. 1.: System composition can be modelled in the form of an upside-down tree.

The terms 'whole', 'system', 'sub-system' and 'part' have no absolute meaning. Parts may be seen as 'wholes' or 'systems' in their own right. Any object, of which a model is made, is part of a larger whole: buildings are part of cities, while cities are part of regions or nations, and so on. On the other side, even the smallest object that is modelled consists of things that are smaller. Hence, no absolute dividing line can be drawn between the model of a system and the context in which it is placed. For this reason, the presented theory does not use the terms 'system' or 'part'. These terms are only used for explanatory reasons, and are therefore placed between parentheses in figure 1.

The present theory is about models of systems. Models are structures of knowledge. The circles in figure 1 refer therefore to Units of Knowledge (UoK's). These Units may comprise any kind of knowledge about any subject. For reasons of comprehensibility, Units of Knowledge will also be referred to as 'knowledge objects', or simply 'objects'.

2.2 System integrity modelling: connectivity networks

Characteristic for a system is that the whole is more than the sum of its constituent parts. The parts of a system complement each other; they interoperate. A single malfunctioning part may cause malfunctioning of the system as a whole.

The interoperability of parts of a system, and thereby the integrity of a system as a whole, can be modelled by means of connectivity networks. Such networks consist of nodes and arcs that connect these nodes; see figure 2.

In the context of the present theory, the arcs between nodes will be called links.

The proposed kind of representation permits only the modelling of bilateral links, as each link involves only two nodes. In practice, more complex interactions may occur. For example, an electromagnetic field that is generated by one system may interact with multiple other systems. In all cases where the theory has been applied so far, it appeared that complex interactions can be reduced to bilateral interactions. Specific cases and exceptions to the rule will be discussed separately in chapter X.

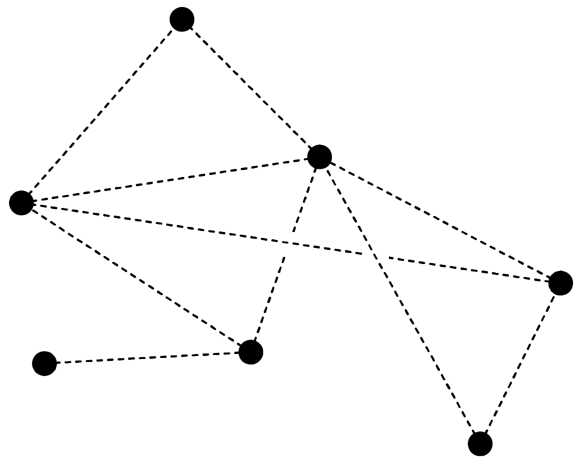


FIG. 2.: A connectivity network consists of nodes (black circles) and arcs that interconnect nodes (dashed lines).

2.3 Compositional structures of connectivity networks

Connectivity networks (chapter 2.2) are orthogonal to compositional hierarchies (chapter 2.1) but will be combined.

For explanatory reasons, the diagrams in this paper use two kinds of views that should also be understood as orthogonal views: a *connectivity view* and a *composition view*.

Figure 3 is an example of a composition view. In this figure, connectivity networks are depicted as black circles connected with dotted lines placed within an ellipse. Knowledge objects are depicted as white circles. An ellipse marks the boundary of a connectivity network that belongs to a higher level object. The black circles represent objects within this network and are connected to lower level objects.

This notation depicts that an object can be seen as a system, which contains a number of interconnected sub-systems. Each subsystem can on its turn be seen as a system, and so on.

A white circle represents the core of a knowledge object. The nodes in a network (black circles) provide additional information about the knowledge object, namely to which other objects it is connected within a particular network.

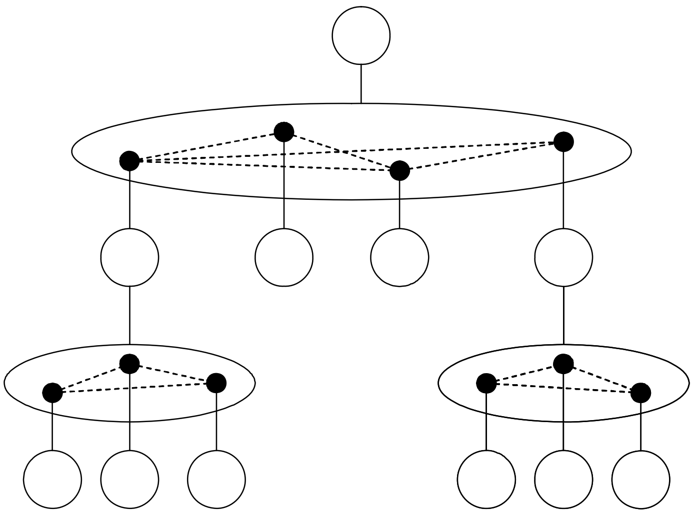


FIG. 3.: Example of compositional structure. Composition view.

2.4 Multi-view connectivity networks

There can be many types of connectivity between objects. For instance, objects may interact from a mechanical, electrical or control perception. Each type of interaction is represented by a different connectivity network. This is depicted in figure 4.

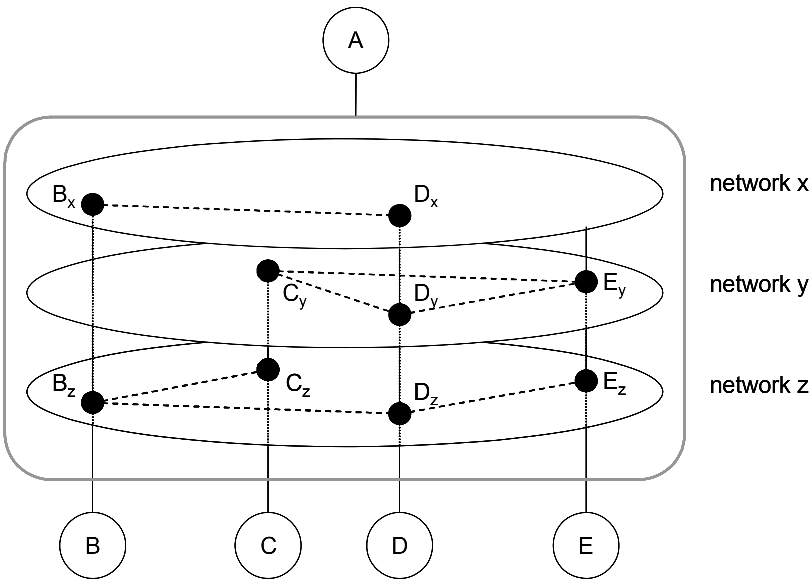


FIG. 4.: Three perception networks. Object B appears only in perception networks x and z. Composition view.

This figure shows that system A has subsystems B, C, D and E. Each subsystem plays a role in one or more connectivity networks. Subsystem C, for example, is connected with D and E in network y, and with B in network z. It plays no role in network x.

All three networks belong to system A, and are therefore shown here in a surrounding box with rounded edges, that is linked with A.

3. MODULAR SYSTEMS

3.1 The need for modular systems and modular knowledge

Although the final design of a single product may result in a fixed compositional hierarchy and a fixed connectivity network, there are many phases in the life of a product where composition and network need to be changed. This happens, for example, during the design process in the examination of alternative solutions, during production in the manufacturing and assembly process, during operation and maintenance in the repair and replacement of components, and during demolition.

A fixed system description is also not useful for products of which the configuration can be changed. A car, for example, has several options built into its design, such as the clients choice between different engine types, manual or automatic transmission, different upholstery, drivers seat on the left or right, and so on.

The flexibility of a product specification will therefore be greatly enhanced by modularity. This has consequences for both the composition hierarchy and the connectivity network.

3.2 Modular composition hierarchies

Modular systems require *slots* in which *modules* can be placed.

In order to permit different engines to be placed in a car, without having serious implications for car body and other systems in a car, the interfaces between engine and other parts should be standardized. If all engines use the same fixtures and have these on the same location, one engine can easily be replaced by another. The whole set of interfaces between system and part will be called *slot*.

Slots are usually designed from the perspective of the system in which a part will be placed. They are based on generic ideas of the modules that can be placed into them. Nice examples of slots can be found in computer systems: a computer has external slots for a keyboard, a video monitor, one or more printers and other peripherals, and internal slots for memory, a graphics card, hard disks, and so on. In order to work with a computer all modules should be installed, but it is possible to replace one module by another that has the same function but possibly with different performance.

This concept can be graphically depicted by splitting the circle, which represents the object in a composition view, into two halves. The upper halve represents the slot in which a module can be placed, and the lower halve the solution offered by another designer or a supplier; see figure 6. This depiction is sometimes referred to as the 'Hamburger notation'.

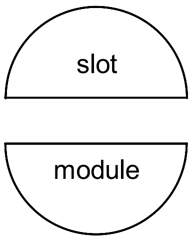


FIG. 5.: Information about an object is split into a slot and a module. Composition view.

A slot contains boundary conditions for a module. Hence, a module cannot be placed in any slot.

For reasons of consistency, all objects in a composition hierarchy are seen as modules within a larger whole. Figure 6 shows the result, if applied to a composition hierarchy. The grey boxes with rounded edges enclose the internals of a module, comprising slots for modules on a lower level and connectivity networks between these slots. In the figures, each slot, indicated with the letter S, has a unique identifier, which is composed of the module name and a sequential number. For example S(A.1) refers to the first slot of module A.

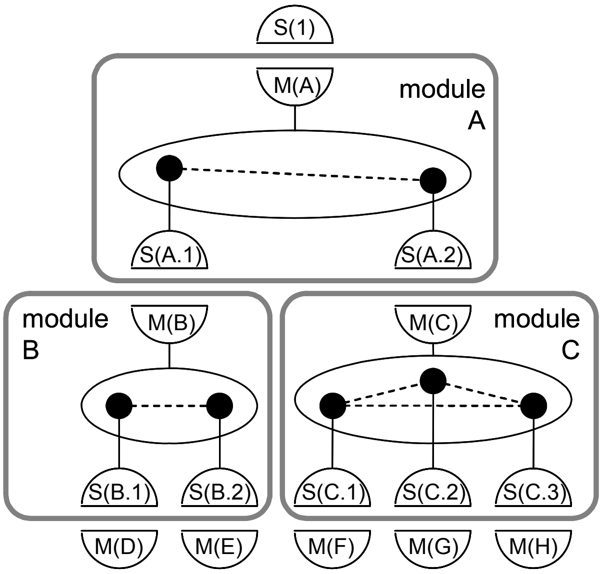


FIG. 6.: Example of a modular composition hierarchy. Composition view.

3.3 Modular connectivity networks: ports and ends

The hierarchical composition structures that reflect systems thinking are - to a degree - artificial. The human brain needs such structures in order to reduce complexity. But in physical reality, anything may be connected with (or interact with) anything else: 'reality is a complex network'.

In the example shown in figure 6, the network of module A contains a link between slots S(A.1) and S(A.2). But it doesn't show how the internals of modules B and C, which are placed in the aforementioned slots, are connected with each other. It may be necessary to obtain also an understanding of connectivity on the lower levels of composition.

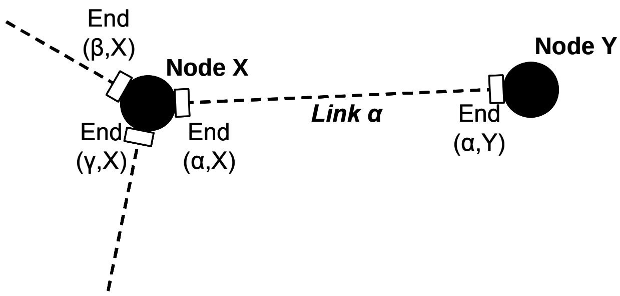


FIG. 7.: Link α ends in Node X via End (α, X). Connectivity view.

Before a solution for this problem can be discussed, the theory of networks has to be extended. Figure 7 shows two nodes in a network, depicted as filled black circles. The link between the two nodes, link α, has two ends: End (α, X) which ends in node X and end (α, Y) which ends in node Y. The ends are depicted as small white rectangles. This figure shows that two more links, β and γ, end in node X.

A node represents the role of an object in a network. By modularizing the system, the object is replaced by a slot for an object. Hence, a node represents the role of a slot in a network. This role may consist of multiple involvements in links with other slots. Each involvement is represented by an End. This is graphically depicted in figure 8.

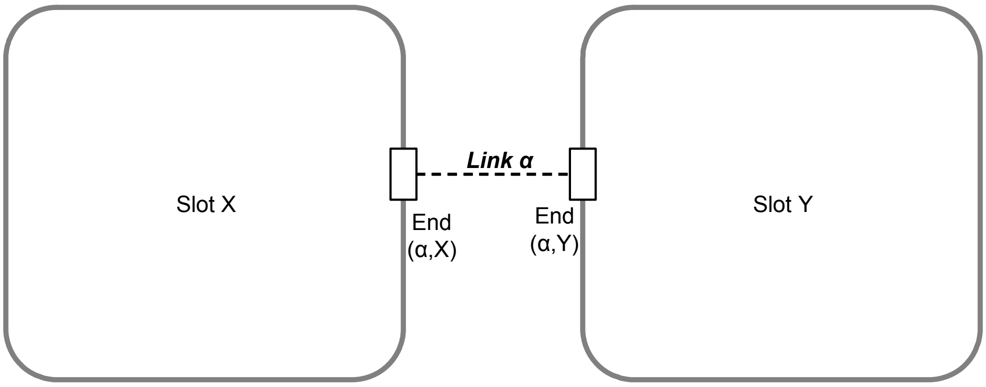


FIG. 8.: Node X denotes the role of Slot X in a connectivity network. Connectivity view.

Figure 9 shows what happens if modules are placed within their slots. Module B is placed in slot X and module C is placed in slot Y. Each module has Ports, depicted here by triangles, that enables it to be connected with other modules. Module B has Port (B,1) and Port (B,2). Node B1 in module B is connected with node C1 in module C via Port (B,1), End (α, X), Link α, End (α,Y) and Port (C,1). Also B3 is connected with C4 via Link α, at the higher level.

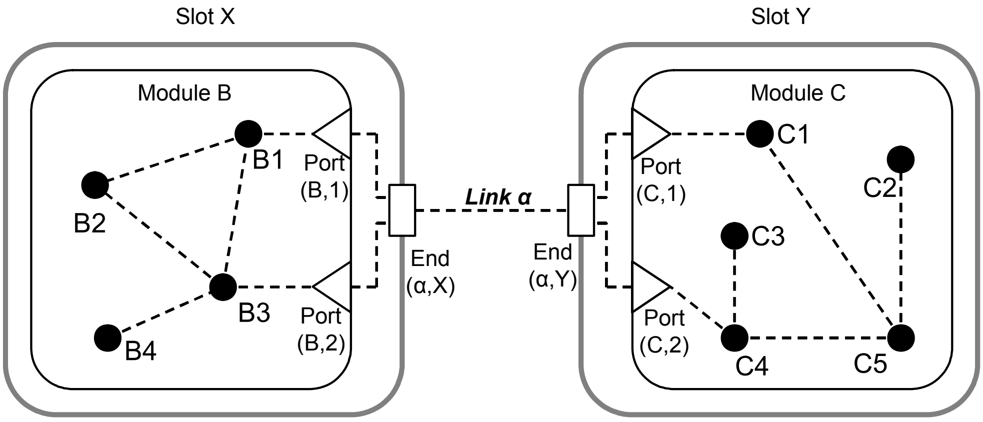


FIG. 9.: Module B is placed in Slot X. Connectivity view.

It is indeed possible that multiple links on a low level of decomposition are interconnected via a single link on a higher level. This may be compared with the pins of a USB connector for a computer. On a global level, the connection between a peripheral device with a computer via a USB cable is a single connection. On a more detailed level it appears that the USB connector has four pins, each of which is connected with different components internal to the computer and the peripheral device. For the connection of ports at lower (i.e. more detailed) levels of composition it is sufficient to apply sequential numbering. In practice, most plugs are designed such that there is only one way to connect them so that the numbers on both sides will correspond. If they are designed to be connected in two or more different ways, the order in which pins are placed can still be relevant, but positioning (i.e. upwards versus downwards counting) should make no difference.

Please note that the example of plugs and connectors is given here metaphorically, just to explain the interconnection of detailed networks via ports. Knowledge about real plugs and connectors, such as the USB connector, need to be described in terms of knowledge modules.

With respect to the diagramming technique, see the following. In diagrams with a connectivity view, slots and modules are depicted as nested rectangles with rounded edges; in diagrams with a composition view they are presented as 'Hamburgers'.

3.4 Selectors

A slot may host zero, one or more modules, but only one at a time. Reversely, a module may be positioned in zero, one or more slots. The interconnection between slots and modules is regulated by *selectors*. Selectors will be depicted in composition schemas as elongated hexagons, placed between the two half circles that make up a 'hamburger'. They form the 'meat' of the 'hamburger'. Figure 10 shows an example with three modules that may be positioned in one slot. As only one module can be placed in one slot a time, the status of the three modules is given below the selector. In this example the first module has the status 'preferred', the second is 'alternative' and the third 'rejected'. There may be other possibilities for the status of a module.

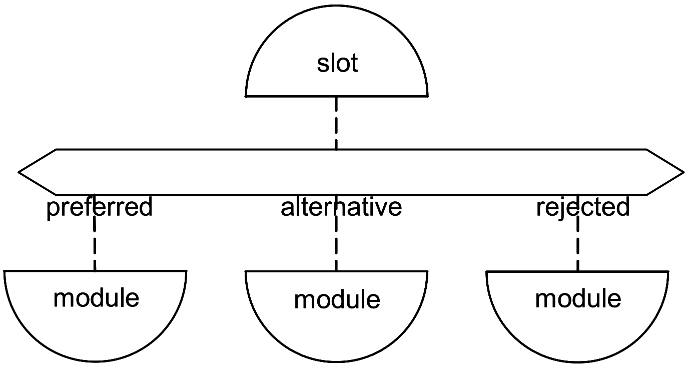


FIG. 10.: A selector is depicted by an elongated hexagon between slot(s) and module(s). Composition view.

The opposite is also possible, namely that a module may be placed in zero, one or more slots; see figure 11. In this general part of the theory there is no constraint on the number of slots in which a module can be placed.

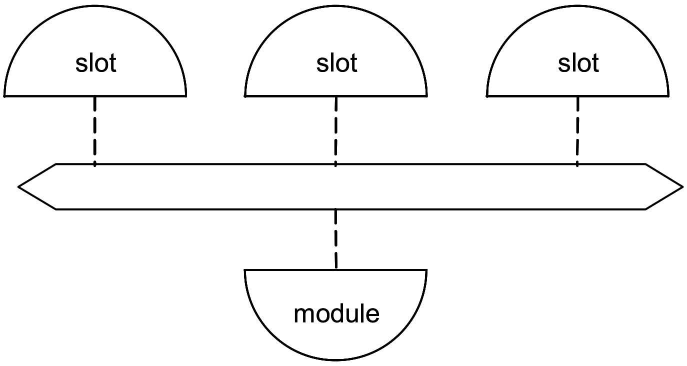


FIG. 11.: This example of a selector shows that a module may fit in more than one slot. Composition view.

4. THE MODELLING OF KNOWLEDGE OF HUMAN ARTEFACTS

This section addresses a number of different principles for the modelling of knowledge of human artefacts. Subsection 4.1 discusses knowledge from existing reality and knowledge that intends to create new reality (including design), 4.2 proposes four different levels for the organization of knowledge, in order to master complexity and redundancy, 4.3 deals with dynamic and static views on reality, including the notion of individual objects and their lifecycle, 4.4 introduces the organization of specifications, and 4.5 addresses generic reusable product and process models, including parametric models.

4.1 The cognitive cycle: from idea to reality and from reality to idea

A distinction is made between:

- *Actual* Knowledge: knowledge from physical reality, such as acquired from observations and experiences, and
- *Imaginary* Knowledge: knowledge that may lead to new reality, such described in design and planning specifications.

All knowledge created in a design process, but also in design analysis, work preparation and planning, is part of the 'imaginary world'. This imaginary world may become reality through production and/or construction. By acquiring knowledge about actual properties and behaviour of an artefact, and by comparing this with the anticipated properties and behaviour, feedback can be given to support a process of continuous improvement. Also the models and algorithms that were used to predict behaviour of the artefact can be improved through such a cognitive cycle (Gielingh 2005); see also figure 12.

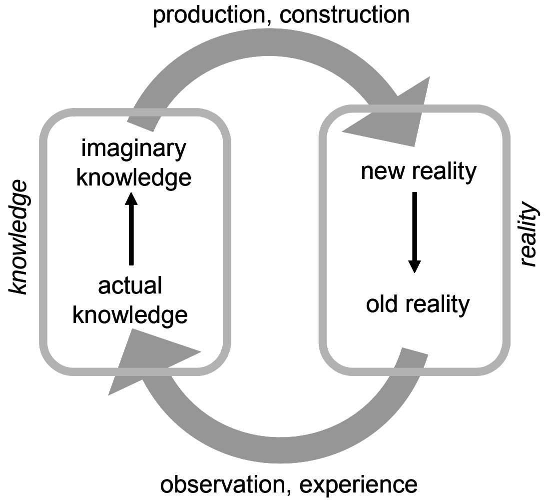


FIG. 12.: The cognitive cycle closes the gap between actual and imaginary knowledge.

4.2 Striving for zero redundant knowledge systems

Reality, including human made artefacts, can be seen as a system of enormous complexity. For reasons of cognitive and structural economy, knowledge about this system - including systems that are part of it - will be dispersed in this theory over four levels, with the aim to reduce or avoid redundant knowledge. Each level can be described as a modular system, such as presented in the previous chapter. But the four levels together form also an interconnected whole and thus also a system.

Suppose that a building which is being designed contains 21 rectangular windows. And suppose that these windows are all the same, except for their dimension: there are twelve windows size 1,2 x 1,8 meter, six windows size 1,8 x 1,8 meter, and three windows size 1,2 x 1,2 meter. It makes sense to design and specify these 21 windows only once. This can be done by defining them in a single parametric model, with width and height as parameters. All knowledge about these windows that is the same can be shared in the parametric model.

Not all knowledge can be contained in the parametric model though. There are twelve individuals that are of size 1,2 x 1,8 meter, and it may be helpful to consider them for production as one batch that has the same specification. Further, each individual will have its own place in the building. Information about their location is different for each individual. Finally, each individual window has its own lifecycle: it is being assembled and installed, it will be maintained and repaired, and it may be removed and installed in another building.

To facilitate the description of all aspects of an object, with minimal redundancy and duplication, the theory identifies four different levels on which knowledge about systems is captured, see also figure 13:

a. Generic level. This level supports the parametric description of products or independent properties, and may therefore be valid for product/property families. A generic description has one or more variables that need to be specified.

b. Specific level. A product or property is specific if it is fully defined and has no unknown variables. A specific description may be used for one or more individuals, for one or more phases in their life.

c. Individual level. At this level, individual objects are recognized for the duration of their entire life. Their properties may however change.

d. Occurrence level. An occurrence corresponds with a particular state of an individual. It provides a static view of an object and is therefore only valid for a certain moment in time. Between two occurrences there is a process of change. On the occurrence level, the specific properties of an object are combined and considered to be ‘frozen’. Also its position is fixed in space and time.

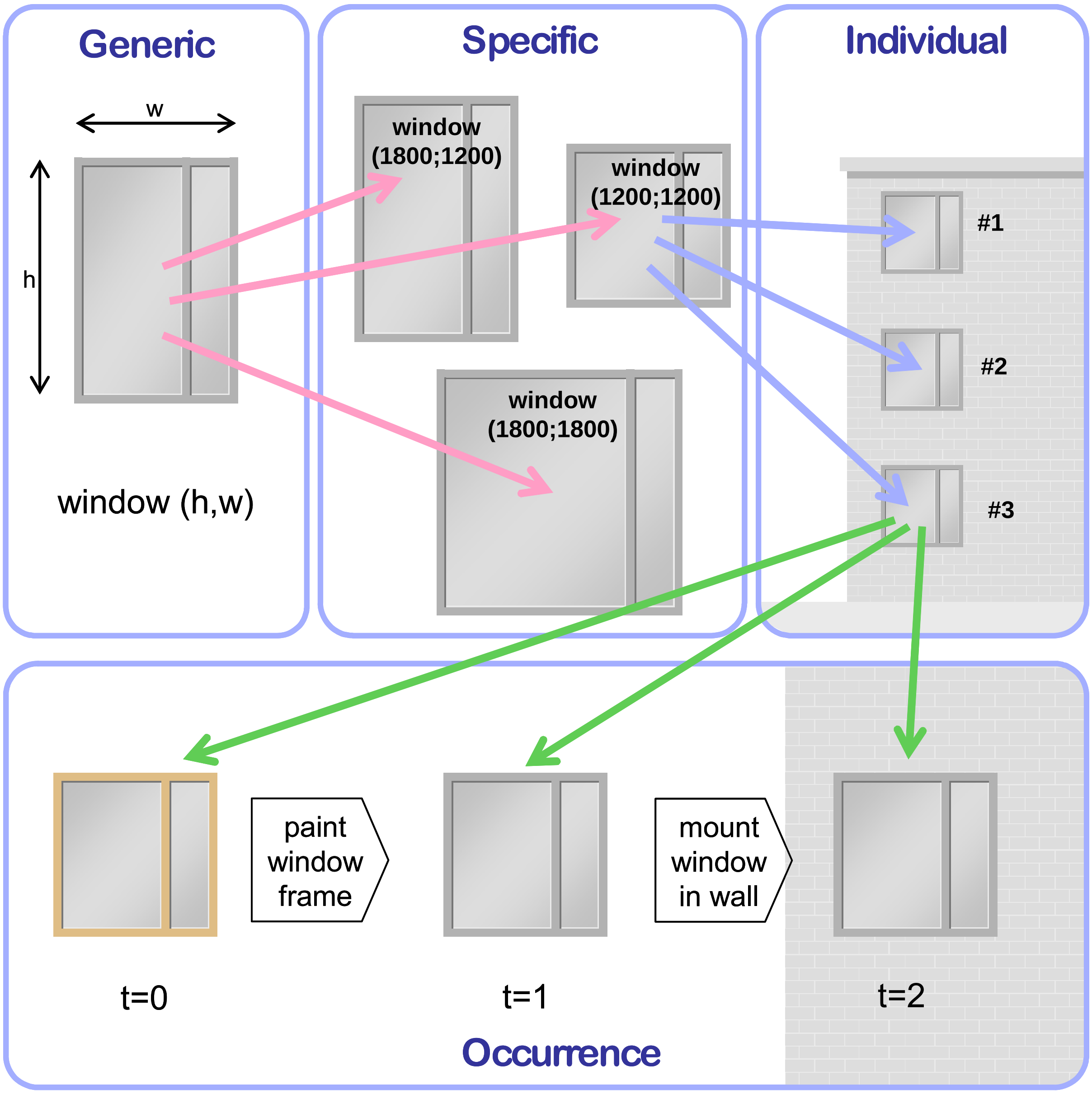


FIG. 13.: The four levels of description, with the example of a window.

Figure 13 gives an example of the window. The generic description has two parameters: height *h* and width *w*. The specification of parameters happens on the specific level. In this example there are three specific windows with dimensions (1800; 1800), (1800; 1200) and (1200; 1200). There are three individuals produced of window (1200; 1200). Each individual goes through phases of its lifecycle; each phase in which properties are stable corresponds with an occurrence. Between occurrences, properties change. Such changes are processes. Hence, processes form the link between occurrences. The example shows only two processes, namely the painting of the window frame and its mounting in a wall.

The generic (parametric) description is not necessarily about an object; it can be about a property or aspect of an object. Shape and material descriptions can be defined separately as parametric knowledge modules, and can be combined on the specific and/or occurrence level. In a design process they are likely to be combined at the specific level, but as properties may change during a lifecycle, new combinations will be formed on the occurrence level. For instance, during production the shape of an object may change, while during operation material properties may change.

The four levels of knowledge are complementary: i.e. knowledge about an object is formed by an assembly of knowledge modules on one or more of the above levels.

During a design phase, this knowledge can be incomplete. In the design process, pieces of information on the Generic and Specific levels are created, and subsequently 'assembled' on the individual and occurrence level. During the life of the object more and other knowledge will be added. This will be discussed in a next chapter.

The four levels do not only apply to static phenomena, such as objects or products, but also to dynamic phenomena, such as activities and processes. Hence, there can be generic (i.e. parametric) descriptions of processes, specific processes that occur multiple times and process occurrences. Process occurrences are shown in figure 12 between object occurrences; they link two different states of a static phenomenon.

Processes do not have to be described independently from static phenomena; they are often logically associated and can then be part of the same knowledge structure. Also this will be discussed in a next chapter.

In the following sections these four levels of description will be examined in more detail, and will be combined with the principles for systems modelling such as described in chapters 2 and 3. As Actual Knowledge is acquired from physical reality, this overview starts with Individuals and Occurrences.

4.3 Occurrence and Individual Level: Static and dynamic views on the lifecycle of individual objects.

Product models often represent a static view on a Universe of Discourse. Reality, however, is in constant change. A static model can only be accurate if it is assumed to be valid for an infinitely small moment in time. A design in construction, for example, describes the intended (and thus imaginary) artefact after production and commissioning, at the moment of handover between contractor and client. It forms the benchmark against which the real artefact can be valued. Designs in traditional construction projects play primarily a legal role.

Reality is in constant change, but many of these changes remain unnoticed, which feeds the impression of a static reality. Mechanical components are subject to wear, chemical alternations (such as corrosion) and material fatigue, which changes their properties. At some point in time these changes may reach a limit, so that the artefact cannot function anymore 'as intended'. Such a change will be noticed, and will often be experienced as a breakdown of the artefact.

Changes occur also during the production of the artefact: material properties and shape change during production.

Changes may be minor, so that they can be interpreted as different phases in the lifecycle of a single individual object. But changes may also be more dramatic, and affect the notion of existence of an individual object. In such a case, the object will terminate its 'life'. Whether or not a change is interpreted as the 'birth' or the 'death' of an individual object is a matter of perception. It is not possible to make absolute statements about 'birth and death'.

The lifecycle of an object can thus be modelled by making a distinction between *individuals* that have a 'finite lifetime' and *static occurrences* that describe an individual for an infinitely small moment in time. Between two subsequent static occurrences there will be a notion of change. This notion of change will be called a *dynamic occurrence*. A dynamic occurrence may also be called a 'process occurrence'.

Figure 14 shows an example for the production process of a vase. This process starts with a lump of clay, which is split into two parts. One part is used to produce the body of the vase; the other is used for the foot. By splitting the lump clay, its lifetime has ended from a discrete parts point of view. The two pieces that remain become therefore two new individuals. These two pieces are reshaped into the vase body and the foot, and are subsequently combined into a 'vase assembly'. This starts the lifetime of a fourth individual object. The vase assembly has its final shape, but is still made of clay. It will therefore be baked, which changes its material properties: the clay becomes pottery. Subsequently it will be painted and covered with a layer of glaze. Again, the vase is baked in an oven, to harden the glaze layer. After this final step, the vase is ready.

In this example, there are seven distinct process occurrences, in which properties of individual objects are changed. In two processes - the splitting of the lump of clay and the assembly of vase body and foot - these changes were such that they result in different individuals. The model of occurrences supports however the traceability of 'origin' and 'future' of these individuals.

Other changes, such as shape changes and material changes, do not affect the nature of the individual. As stated above, whether a change affects the nature of an individual is a matter of perception. In many cases it is its *function* which determines its lifetime. In the example of the vase, we may state that it only becomes a vase after its properties are such that it can function as a vase. In that case the vase starts its life after the production is finished, in step 4.4. The individual that undergoes the production process from assembly to finishing should then be called a 'pre-vase'.

The model in the middle of figure 14 shows static occurrences, depicted as ovals, and dynamic occurrences (or process occurrences) depicted as rectangles. The model on the lower side shows the individuals as horizontal lines and the static occurrences as black dots (1.1, 2.1, 2.2, etc). The occurrences are identified by an individual id. (digits before the dot) and by an occurrence id. (digits after the dot).

For each static occurrence it is possible to provide a static model of the object. To avoid redundancy in the product model, specifications of Imaginary Properties are given at the Specific Level of the product description, and these may be independent of each other. Thus, for example, the shape of the vase can be described only once at the specific level, and be referenced by static occurrences 4.1, 4.2, 4.3 and 4.4. The material properties of the unbaked clay may also be described only once at the Specific Level, and be referenced by static occurrences 1.1, 2.1, 2.2, 3.1, 3.2 and 4.1.

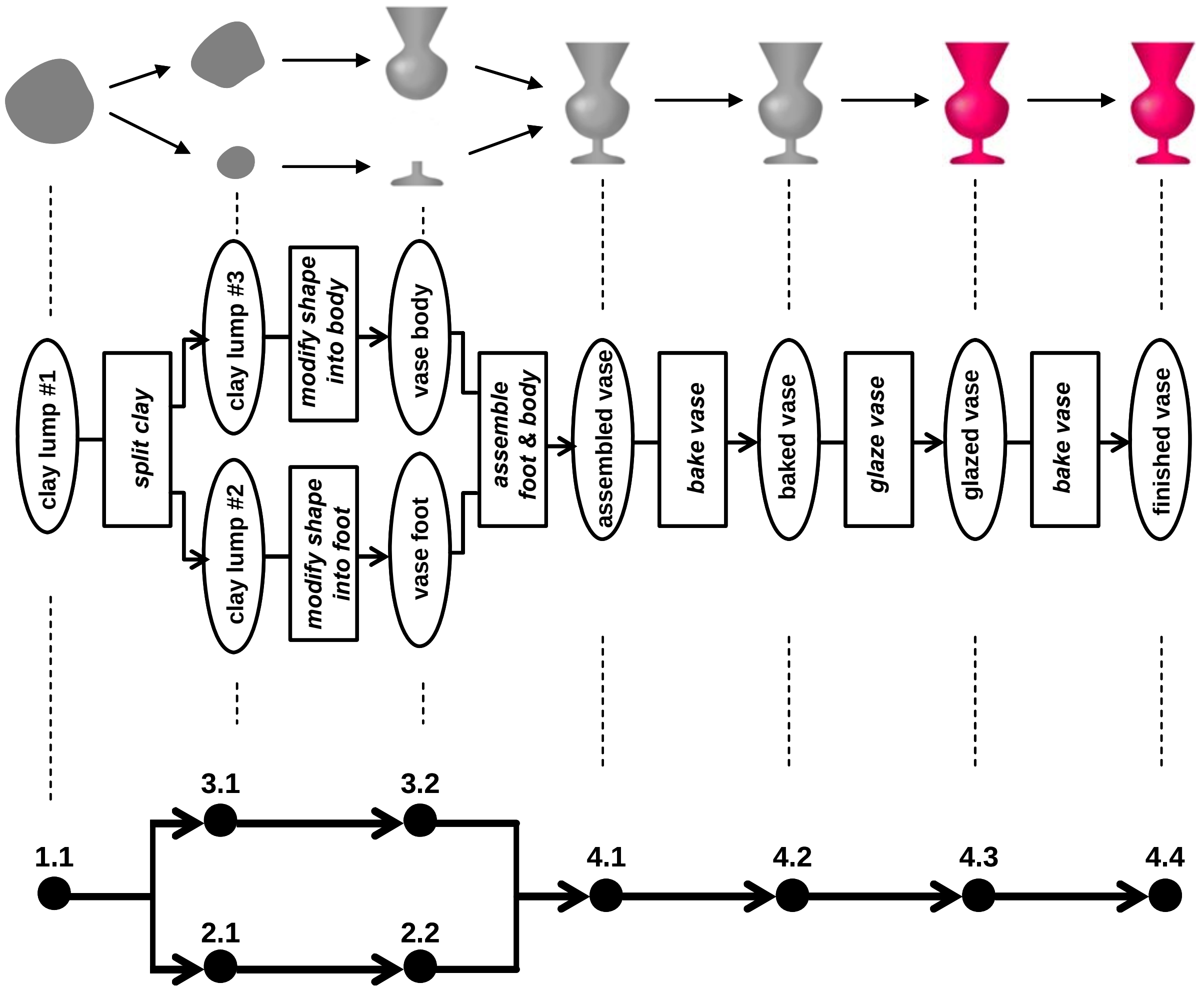


FIG. 14. Example of the production process of a vase, modelled with static and dynamic occurrences.

Dynamic Occurrences contain knowledge about their interconnection with Static Occurrences. Hence, the dynamic occurrence 'assemble foot and body' refers to the static occurrences that form its input (2.2 and 3.2) and output (4.1).

Occurrences contain also knowledge about location in space and location in time.

Models can be developed for Imaginary and/or Actual Individuals and Occurrences. The design and production process planning of the vase results in an Imaginary model. As stated above, such property specifications can be shared by multiple occurrences and are therefore placed on the Specific Level.

Measured Actual Shapes are Actual Properties, and stay on the Occurrence level. They permit the tracking and tracing of actual property changes of an artefact during its lifetime.

4.4 Occurrence and Individual Level: The Installation Hierarchy

The physical configuration of an artefact is described with an *Installation Hierarchy*, using the principles outlined in sections 2 and 3. An Installation Hierarchy obeys additional rules that will be discussed in this section.

The modules in an Installation Hierarchy represent individuals. Each module corresponds with a *technical individual*, which can be understood as a *physical object*. The role of this object in a larger whole is defined by the *functional individual*. The 'hamburgers' in an Installation Hierarchy are thus formed by Functional and Technical Individuals.

Functional and Technical individuals may have different lifetimes. If a technical individual is subject to wear and tear, its lifetime may be shorter than that of its function. In such a case it has to be replaced by another technical individual. The reverse may also happen: the technical individual may have a longer lifetime that its functional place in a larger system.

An example is the tyre of a car. As a car cannot drive without tyres, the functional lifetime of its tyres is as long as the lifetime of the car. The technical lifetime of tyres depends on many factors, but is usually much shorter than the lifetime of the car. Worn tyres will therefore have to be replaced by new ones. But now suppose that a 10 year old car breaks and is declared a total loss. Suppose that new tyres were installed a few months before, and that these tyres can still be used for at least one more year. In that case the tyres can be re-installed on a different car. This is an example of a (physical) configuration change.

A single association between a functional and a technical individual is called an *installation occurrence*.

A different example may clarify the usage of installation occurrences. Suppose that there is a plant in which two pumps A and B modify the pressure of a stream. From the perspective of the whole (i.e. the plant) the pumps are functional individuals. Functional Individuals appear on Process Diagrams. During operation, pump A may be used more frequently than pump B. After 4 years of operation, pump A may have run for 16.000 hours while B has run only for 4.000 hours. Suppose that similar technical pumps are installed for A and B, having an expected lifetime (without refurbishment) of at least 20.000 hours. Suppose also that a major maintenance stop happens every 2 years. If the pump installed at location A is kept there, it may break before the next planned maintenance stop. Therefore the maintenance manager decides to swap the physical (technical) pumps, so that after 4 more years of operation both will have ran for 20.000 hours. Then, after 8 years of total operation and during a planned (regular) maintenance stop, both pumps can be replaced.

This change of (physical) configuration can be modelled with the Installation Hierarchy, see figure 15.

Two types of Installation Hierarchies will be used in practice:

(1) The *Imaginary Installation Hierarchy* describes a physical artefact before actual changes to its configuration are made. It is used for maintenance and/or operational planning or simulation.

(2) The *Actual Installation Hierarchy* describes a physical artefact 'as-is'. It is used for the collection of operational data, maintenance inspection data and other data that describe state and condition of a system and its components.

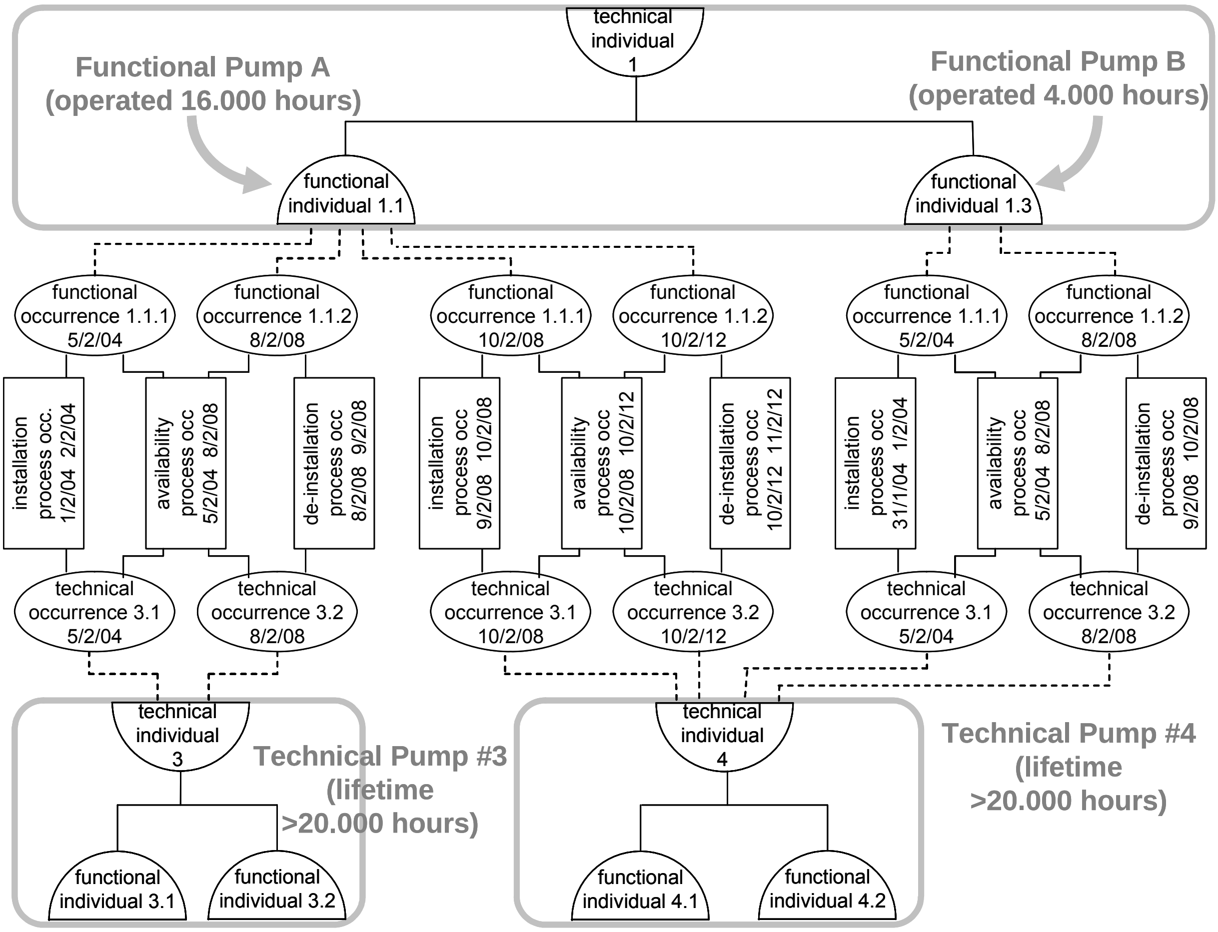


FIG. 15. Links between modules in an Installation Hierarchy describe configuration changes of a system.

4.5 The Specific Level: Product and Process Specifications

The Specific Level is used for the specification of products and processes, such as in design. It contains descriptions of objects and processes - or properties of objects and processes - that can be shared by individuals and/or occurrences. These descriptions are fully specified: they are non-variable and thus non-parametric. Only location in space and time remains unspecified; this information is present on the Occurrence Level.

In the example given in figure 14, the shape of the vase can be shared by static occurrences 4.1, 4.2, 4.3 and 4.4. To avoid redundant information, this shape will therefore be specified only once and resides on the Specific Level. Similarly, the material properties of unbaked clay can be shared by static occurrences 1.1, 2.1, 2.2, 3.1, 3.2 and 4.1, while the material properties of pottery are shared by occurrences 4.2, 4.3 and 4.4.

Also on this level, two types of descriptions are recognized: *Imaginary Specifications*, which describe products and/or processes before they physically exist, and *Actual Specifications*, which describe products and processes 'as-is'. An Imaginary Product Specification is similar to a product design, and an Imaginary Process Specification is similar to a process plan.

As design and planning play an important role in product development, only Imaginary Specifications will be elaborated in this section. Actual Specifications will be discussed in section XXX.

4.5.1 Imaginary Product and Process Specifications

Imaginary Product and Process Specifications describe artefacts that do not (yet) exist. Such specifications comprise knowledge captured in traditional design, but also in requirement specifications, procurement, work preparation and process planning.

First it will be discussed how a *design* is represented.

A design process aims at the *specification* of a new product and/or process. A design specification can be seen as a *system*, in line with the theory that has been discussed in chapters 2 and 3. It contains global specifications and detailed specifications that form a composition hierarchy.

A design contains also an intricate network of relations, associations and dependencies. If a design is organized as a modular system, the design process can be interpreted as the configuration of technical solutions that, together, meet the higher level goal that is set at the beginning of the process.

The Specific layer consists of two types of specifications: *implicit* and *explicit* specifications. An implicit specification refers to a generic (i.e. parametric) knowledge module and specifies its parameter values. In case of the window in figure 12, the specific module contains a reference to the generic (parametric) window and specifies its parameters. For example: height = 1200 mm, width = 1200 mm.

Based on this implicit specification, explicit data about the object can be automatically generated. In the example this may be a drawing or 3D model of the window; see also figure 16. Such explicit data are redundant with the parametric specification: one is derived from the other. Therefore the combined implicit specification and the parametric description will be called a *definition*, while the explicit data are called *derivations*.

It is also possible that a specific module does not make use of a generic, parametric module. The designer that creates such a module will then directly provide explicit data, such as an explicit shape and material definition. Non-parametric modelling is still the most common way of designing. Other explicit models may be derived from such an *explicit definition*.

A third option is to consider the procedure that resulted in a model as a definition. In geometric modelling, constructive solid geometry (CSG) is a combination of parametric modelling (through parameter definition of elementary shapes) and procedural modelling (through Boolean operators).

A definition is often referred to as 'design intent' (Kim et al 2008).

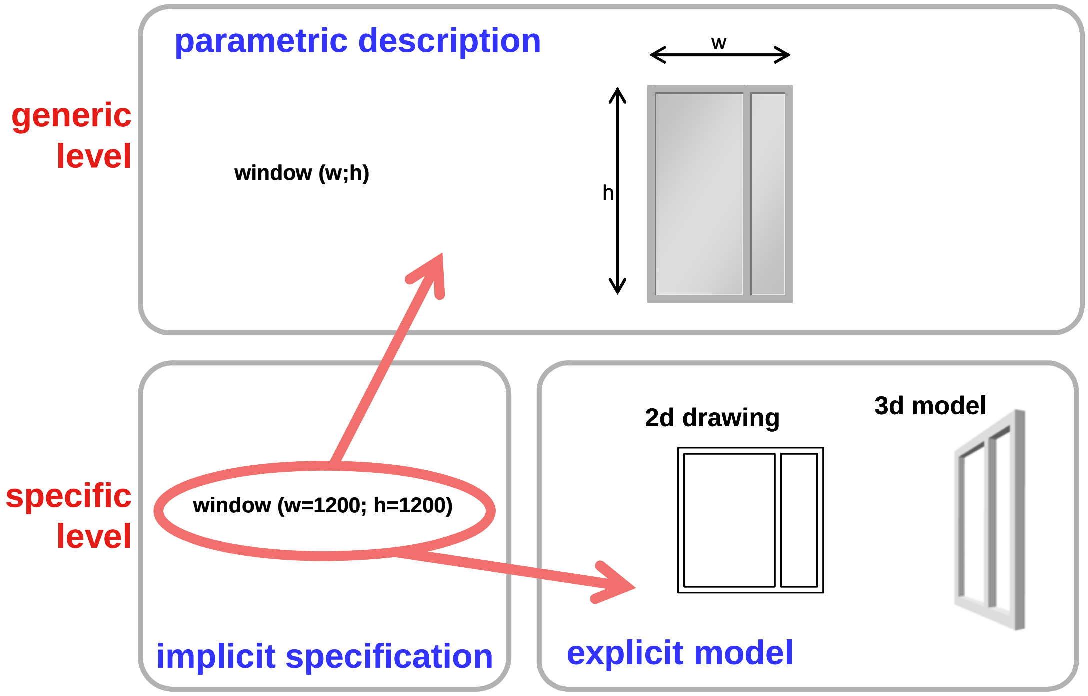


FIG. 16. An implicit product definition refers to a parametric model and defines parameter values.

The distinction between *Definitions* and *Derivations* is important for product and process modelling: it makes clear which knowledge modules form the 'master' and which are 'copies' that are derived from the master. Changes should always be made in Definitions (the master modules). Derivations (copies) should be updated accordingly.

Combinations of implicit, explicit and/or procedural specifications are possible. In such cases, use is made of generic, parametric modules, but the explicit derivations may then be augmented or modified by the designer. In such cases the modifications can be archived and used as input for the creation of new generic, parametric modules, possibly as *variants* of existing modules. Not the end-result, but the procedure that led to the modification becomes part of the new parametric description.

4.5.2 The Specification Hierarchy

A human artefact is a system, but also the specification of this artefact can be seen as a system. Therefore the principles outlined in chapters 2 and 3 apply also to Product and Process Specifications. The resulting composition hierarchy differs on essential points from an Installation Hierarchy.

To emphasize this difference, the hierarchy will be called a *Specification Hierarchy*. Slots will be called *Functional Specifications* (or, alternatively, *Functional Units*, in accordance with the older GARM theory) and modules will be called *Technical Specifications* (or, alternatively, *Technical Solutions*).

A Functional Specification defines the role of a subsystem or part in a larger whole. It defines also requirements and boundary conditions for the subsystem or part. In the course of an unfinished design process, it may also be considered as a design problem.

A Technical Specification describes the subsystem or part, independent of its role in the larger whole. In the course of a design process, it can also be seen as a solution for a design problem.

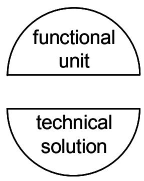


FIG. 17. Functional Specifications and Technical Specifications complement each other.

A Technical Specification Module contains a full specification of an artifact and/or process on a particular level of (de)composition. It contains also Functional Specifications of subsystems or parts, but not the technical details of these subsystems or parts. This principle keeps a specification modular.

One important difference between a Specification and an Installation Hierarchy is that each Specification Module may refer to many individuals, whereas an Installation Module refers to a single individual.

But, where an Installation Hierarchy is a structure that represents a product and/or process, a Specification Hierarchy is first of all a modular structure of *knowledge*. It carries all implicit and explicit knowledge that is shared by zero, one or more individuals and/or occurrences.

Figure 18 shows how Functional and Technical Specifications are linked. For each ‘design problem’ there can be zero, one or more ‘solutions’. Multiple solutions can be evaluated as alternatives. At the end, only one will be selected; others will be rejected. Decision arguments that lead to selection or rejection can be associated with the selectors.

Knowledge modules are potentially reusable. Figure 18 shows that solution 4, which was proposed as a solution for problem 1.3 but rejected, can be an acceptable solution for problem 2.3.

In a design process, solutions can be evaluated in the context of the larger whole in which they are placed. The results of such evaluations are associated with the specification modules.

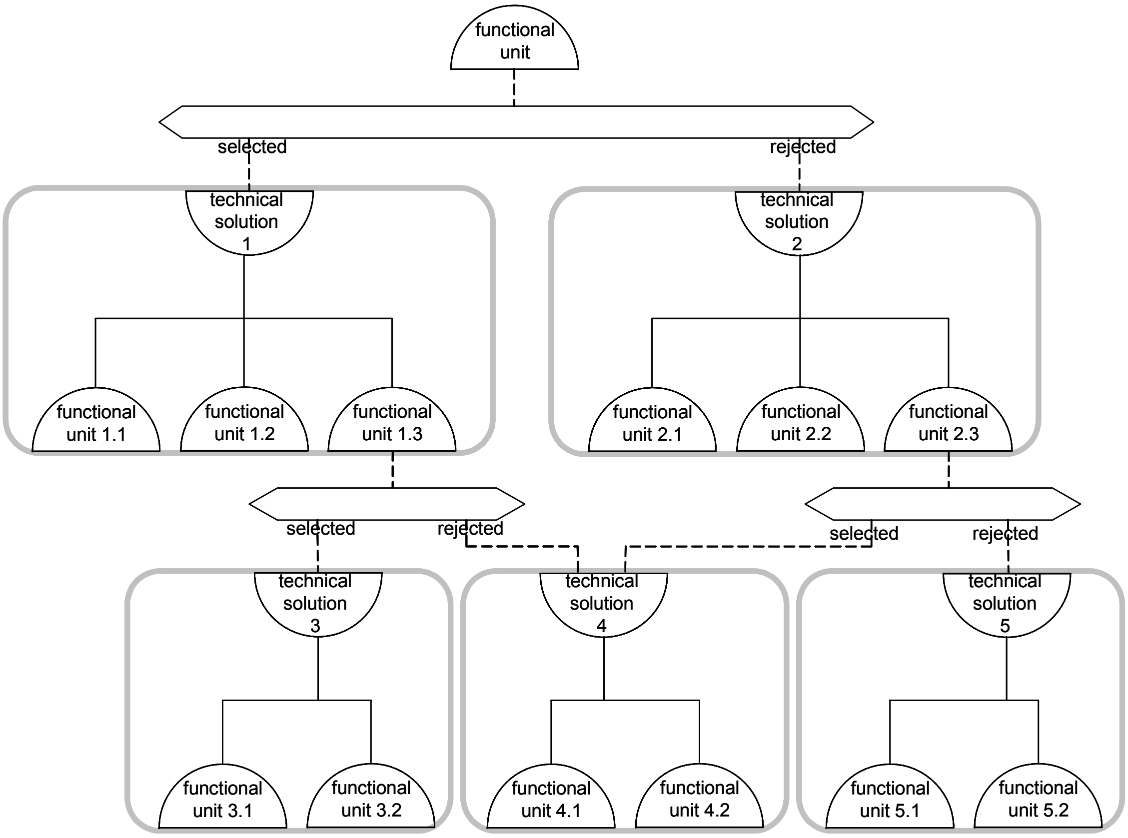


FIG. 18. An implicit product definition refers to a parametric model and defines parameter values.

4.6 Generic level: parametric models of design and lifecycle knowledge

The generic level consists of descriptions of knowledge which are not fully specified but have variables. Such descriptions may refer to families of objects and can be mathematically expressed as parametric objects. Examples of parametric object descriptions are features (such as used for the design for mechanical parts), knowledge frames and ontologies.

Parametric descriptions of knowledge objects, such as products, processes, problems and solutions, may form conceptual hierarchies, also known as generalization/specialization hierarchies. The structure of hierarchies is determined by parameter-values.

An example of a parametric description is that of a parallelogram. This shape can be specified with the three parameters *base*, *altitude* and *angle*. A parallelogram remains a generic (variable) shape as long as one of the parameters is not defined.

As stated above, parametric descriptions can be part of a conceptual hierarchy. This applies also to geometric shapes. A square is a special kind of rectangle, a rectangle is a special kind of parallelogram, a parallelogram is a special kind of trapezium, and a trapezium is a special kind of quadrangle. All these shapes have four corners which are connected by straight lines. In addition, a trapezium has two parallel sides and a parallelogram two pairs of parallel sides. A rectangle is a parallelogram with four rectangular corners, and a square is rectangle of which the sides have the same length. A square requires only one parameter to be defined: the side length. A rectangle requires two parameters, and a parallelogram three. The more generic the shape is, the more parameters it requires. But at the end it is possible to define a square in terms of a quadrangle.

Taxonomies (i.e. classifications of things) and ontologies (i.e. categorizations of things that exist) follow a similar pattern. Ontologies associate knowledge with classes of things. The more generic concepts in an ontology are, the less knowledge they embody, and the larger the population of things to which they refer. A special concept can be defined in terms of a more generic concept; for example: a square is a special kind of rectangle, and a rectangle is a special kind of parallelogram.

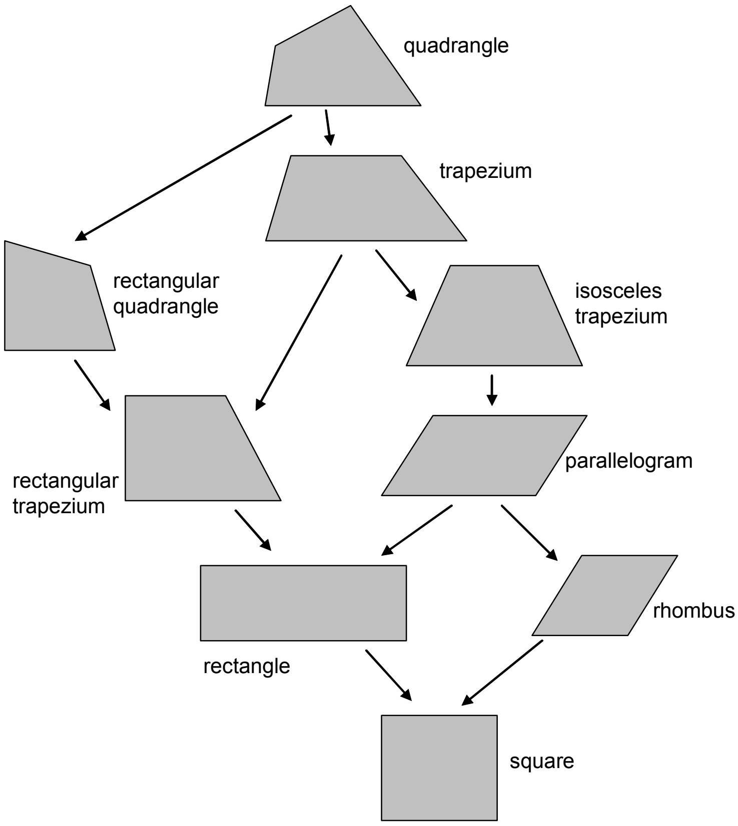


FIG. 19.: Example of conceptual hierarchy of quadrangles.

Conceptual hierarchies are not restricted to tree-shaped hierarchies; they may form lattice-shaped structures, such as shown in figure 13. This example shows that a square can be seen as a kind of rectangle, but also as a kind of rhombus.

While descending from generic to specific in a conceptual hierarchy, an increasing number of parameter values will be defined or constrained. Hence, the number of variables decreases. A square has only one variable. Fixed parameter values become part of the *intentional* definition of concepts in a hierarchy. For example, a square can be intentionally defined as a parallelogram with an angle of 90^{o} of which base and altitude are equal.

For the formation of conceptual hierarchies, it is possible to apply the principles that were outlined in chapters 2 and 3. Hence, it is possible to create *modular* conceptual hierarchies. Seitz and Kirkley (1991) proposed such a modular structure for ISO 10303 (Standard for the Exchange of Product Model Data); see also (Gielingh 1992). More recently, initiatives are taken to apply modular principles to this standard (Kemmerer 1999).

Placement of parametric descriptions in a conceptual hierarchy is preferred but not necessary.

A specific example of parametric description is that of *form features*. Feature technology is a well developed sub-domain of geometric modelling technology, used in particular by the manufacturing industry. For an overview of relevant publications in this field see (Pratt 2007).

4.7 Process modelling

In the discussion of generalization and specification hierarchies, no distinction is made between product and process descriptions. It appears that the same phenomenon can be viewed as being static or dynamic. In fact, all natural phenomena are subject to change, and are thus dynamic. To consider a phenomenon as being static is therefore a matter of perception. During an infinitely small piece of time, all phenomena will look static.

The theory presented here makes therefore no fundamental distinction between static and dynamic aspects of phenomena. Only by taking an infinitely small time-slice, which corresponds in this theory with an occurrence, a phenomenon will be seen as static. During a longer period of time, one or more aspects of the phenomenon may be subject of change.

Certain 'properties' of phenomena play a role in their classification or typification. They are part of their intentional definition. If these 'properties' change, the nature of the phenomenon changes. This terminates the lifetime of the phenomenon, and thus its existence as an individual.

The whole of 'properties' that is associated with a phenomenon, including 'properties' that form its intentional definition (and thus its classification), is called 'object DNA' and has to be associated with the phenomenon. This part of the theory is introduced in (Gielingh 2005) and elaborated further in (Gielingh 2008a). In the latter publications, 'properties' are called 'notions', which explains why the term "property" is placed between single quotes here.

Generic and Specific descriptions may not refer to all 'properties' of a phenomenon, but just to one aspect. If they refer to an aspect of change, they will be classified as processes. Phenomena, being aggregates of 'properties', cannot be classified as being either static or dynamic: they always combine both aspects.

4.8 The knowledge lifecycle

The previous sections discussed in more detail three kinds of hierarchies - installation, specification and generalization hierarchies - and the distinction between imaginary and actual knowledge. If combined, they result in six kinds of hierarchies, as depicted by figure 20.

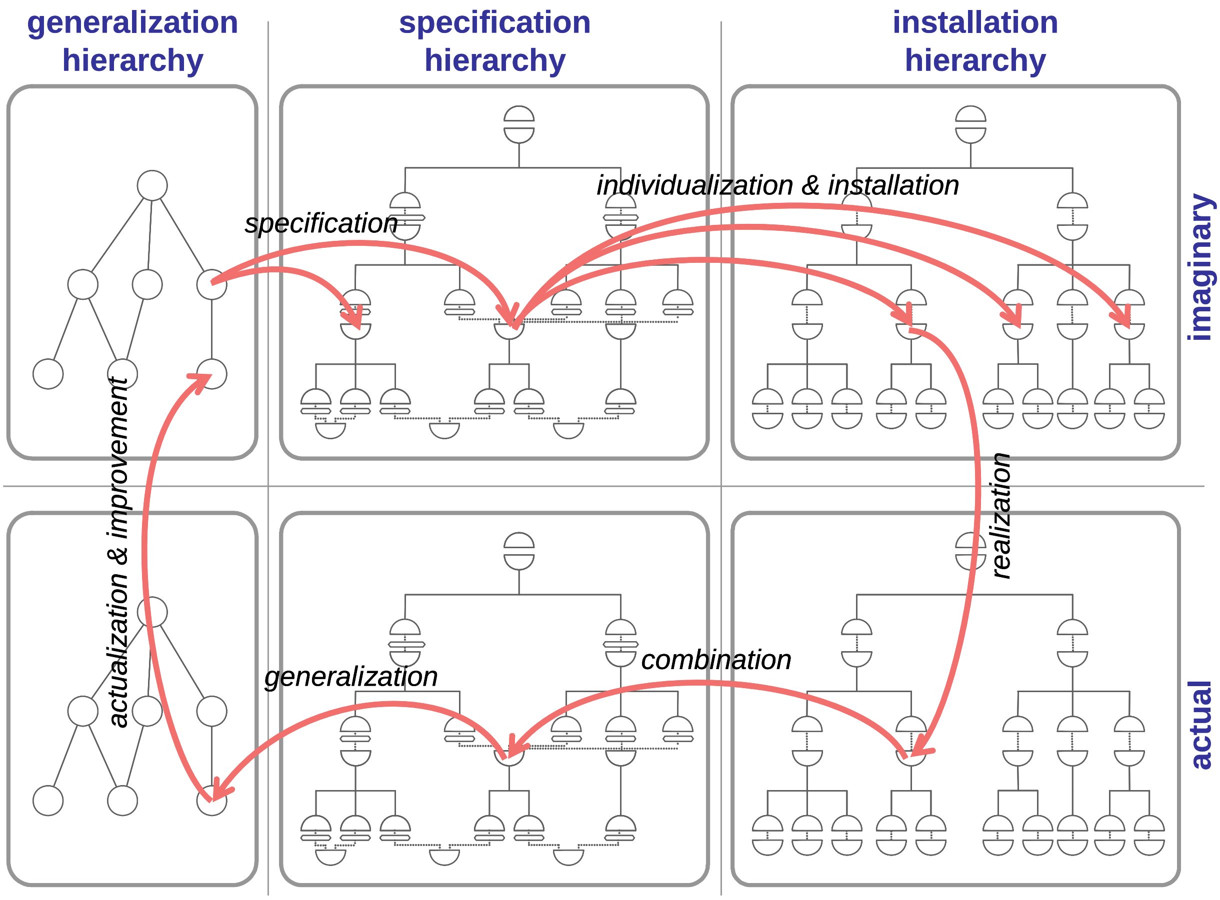


FIG. 20.:Role of the six resulting hierarchies in the product/process knowledge lifecycle

A design process aims at the production of an imaginary specification hierarchy, for which use can be made of generic reusable knowledge modules that are part of a generalization hierarchy. Once the specification is ready, the individual components can be made explicit in an imaginary installation hierarchy. The conversion of a specification hierarchy into an installation hierarchy can be automatic. For each individual component, individual processes can be defined, for example for project planning and scheduling, commissioning, operation or maintenance. This results in component occurrences: descriptions of the component in different phases of its lifecycle.

Once the designed artefact becomes reality, actual knowledge can be collected and associated with occurrences or processes. This actual knowledge can be compared with the original imaginary knowledge, so that differences between planned (or expected) and actual behaviour become apparent. Knowledge over a large number of individuals and occurrences can be statistically processed and combined, and attributed to specific objects in an actual specification hierarchy. In a subsequent step, this knowledge can be generalized, and associated with generic parametric knowledge modules. These generic knowledge modules refer back to the knowledge modules used for design.

Hence, by linking all knowledge modules in the various hierarchies, a principle of knowledge tracking and tracing can be applied. This results in a *closed loop knowledge lifecycle*, where knowledge modules used for design are continuously updated and improved with knowledge acquired in actual production and use.

For recapitalization, the terms used for slots and modules on the different levels are given below.

level	**slot**	**module**
generic	Generic (Parametric) Specification	
specific	Functional Unit / Specification	Technical Solution / Specification
individual	Functional Individual	Technical Individual
occurrence	Functional Occurrence	Technical Occurrence

4.10 'Extended Design'

The contents of an Imaginary Specification Hierarchy comprise the scope of a design in the traditional sense. However, it will contain substantially *more* knowledge than captured in current design.

A traditional design usually specifies an artefact for only *one* phase of its lifecycle: after production or construction is completed, and before operation and maintenance starts. The Specification and Installation Hierarchies are intended to specify *all* stages of a lifecycle, including stages during production or construction, and stages during operation and maintenance. Hence, it contains also specifications that are needed for work preparation, including temporary structures or results such as ditches for cabling, scaffolds, temporary access roads and storage facilities.

Further, an Imaginary Specification does not stop at the point where procurement starts. Details of components or systems delivered by suppliers can be attached to a design specification and become part of the Specification Hierarchy, so that more complete configuration knowledge becomes available. The attachment of such knowledge requires careful handling with respect to Intellectual Property Rights.

4.11 Knowledge Repositories and Data Warehousing

Knowledge modules created on all levels in all hierarchies can be archived for later use. Also knowledge modules that were rejected in a particular project can be archived for possible future use. The knowledge modules created in design and planning become part of a knowledge repository and/or data warehouse that can be consulted at any time for new projects.

5. A SPECIFIC NETWORK: TOPOLOGY AND GENERALIZED TOPOLOGY

The type of network that is most commonly used in design deals with shape and the topological relations between geometric elements. In construction, different shape models may have different meanings, for example caused by different levels of *shape idealization*. This section discusses how a shape model can be interpreted as a special kind of network, and subsequently be integrated within the hierarchical models. This enables shape decomposition and the modelling of complex shapes using shape modules.

5.1 Perception of physical reality

From the viewpoint of mechanical physics, objects produced by the construction sector consist of matter and voids. Voids – such as spaces in a building - have often functional significance. But for production, the constitution and shape of matter is of primary importance. It is for this reason that an important part of a design process is concerned with the constitution, shape and arrangement of material objects.

5.2 Perceptive models of discrete parts

The expression of the shape of material objects is not trivial. According to (Gielingh, Suhm 93) only two concepts are semantically relevant for product models of material objects: *material volume* and *material skin*. In classic mechanical thinking, material volume is a region of space that is occupied by matter. The typical material properties of homogeneous materials are usually independent of their location inside the material volume. In such cases, material properties do not deviate from nominal values within a defined tolerance zone. Non-typical properties, such as stress, pressure and temperature can vary inside the volume. Such properties are usually expressed with the finite element method.

Material skin is a transition zone of space where material properties change significantly: their deviation from nominal values are beyond a defined tolerance zone. In most cases, it is a transition between matter with one set of properties to matter with another set of properties. Often it is the molecular composition of matter that changes near a boundary, but it is also possible that other properties mark a boundary, such as the state of aggregation, blends with other materials, or structural integrity. In many cases, material boundary is a transition between matter and void.

There are also cases possible where properties are not constant within a volume, so that no nominal values and no clear boundaries can be identified. There are ways to represent such cases, but they will not be discussed in the context of the current theory.

A mathematical way of expressing shape is through geometry. Classical Euclidian geometry can be expressed in terms of algebraic functions according to the theory of Descartes. Points in 1-, 2- or 3-dimensional space are represented as vectors. Algebraic functions enable the expression of curves, surfaces and volumetric shapes. If a volumetric shape cannot be expressed directly in terms of a single geometric primitive, it is possible to extend the vocabulary of geometric shapes with transformations of solid primitives - such as sweeping, tweaking or deformation) - Boolean operations on primitives - constructive solid geometry - and/or the formation of a closed surface by combining a number of analytic or free-form surfaces. The latter approach forms the basis for one of the most popular expressions of geometric shapes for computer aided design: boundary representation.

A complex shape is defined in boundary representation by a set of faces which form together a closed shell. Each face is associated with the geometric description of a surface, and is bound by a closed set of edges. Each edge connects two faces. On its turn, each edge is associated with a geometric curve, and is bound by two vertices. The location of a vertex is defined by a point in 3-dimensional Cartesian space. The mathematical foundation for the connectivity of geometric elements – called *topology* - is developed in the 18th century by the Swiss mathematician Leonhard Euler.

Vertices and edges have no real meaning for the description of material objects. If they are used for the description of geometric shapes in boundary representation, they mark transition zones on the skin where curvature changes abruptly.

The skin of a material object can be considered in such a case as a 2-dimensional space which is orthogonal to the dimension in which strong discontinuities in material properties are noted. An edge marks a 1-dimensional zone of this skin where curvature changes abruptly in the other orthogonal dimension. A vertex marks a 0-dimensional zone where curvature changes abruptly in both dimensions of the skin. In both cases, topologies get a geometric meaning.

Figure 21 shows that a simple material object such as a dice can be described in various ways. In the left hand representation the curvature at the edges of the dice is ignored so that the object is approximated by a cube. The blend information can still be included in such a model, in the form of a parametric, non-evaluated description. In the right hand representation the curvature at the edges is not ignored and expressed explicitly. This means that in boundary representation, the same object can be expressed differently in terms of topology and geometry. The edges and corners of the dice are represented by topological edges and vertices in the left hand model, while they are represented by topological faces in the right hand model.

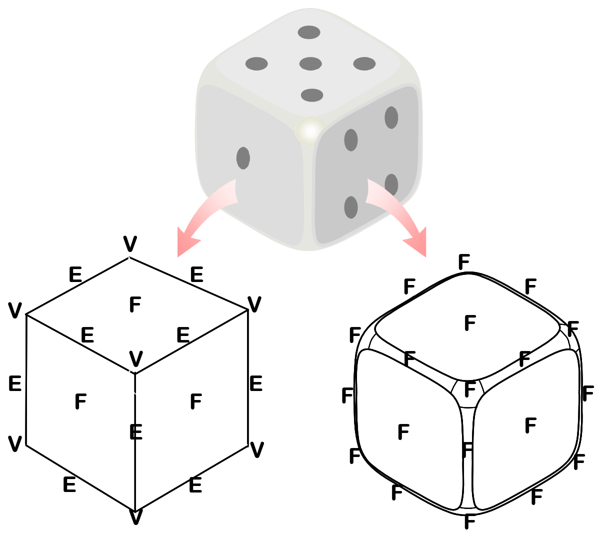


FIG. 21.:Topological models of a dice, expressed at different levels of idealization.

A close inspection of the dice reveals that its sides are not perfectly flat, and that the edges are not perfectly cylindrical: they are to a degree irregular. The boundaries of material objects are never 'as perfect' as a mathematical model.

Geometric descriptions of objects define in most cases a *nominal shape*. Actual shape is irregular and deviates from nominal shape. It is possible to define a minimal shape (which is located entirely inside the material volume), and a maximal shape (located outside the material volume). The space between minimal and maximal shape is a tolerance zone where the actual boundary is located; see also Figure 22.

The tolerance zone between minimal and maximal shape for a 3-dimensional object is a volume. This implies that the sides, edges and corners of the dice shown in Figure 21 may also be represented by topological volumes in stead of faces.

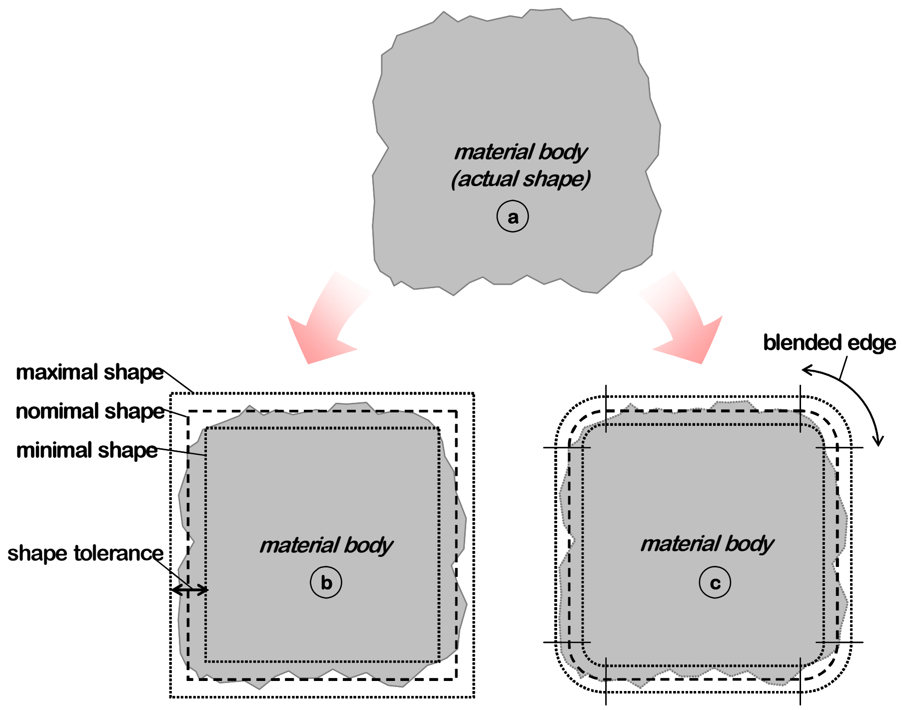


FIG. 22.:Cross-section of the dice (grey area) with exaggerated surface irregularities..

There are apparently multiple ways to describe a simple object such as a dice; in some cases the cube-description (Figure 21 left) will be sufficient, but others may require explicit expression blended edges and corners (Figure

21 right) of nominal shape, and again others may require even a volumetric description of tolerance zones. Furthermore, certain applications may require information about surface roughness.

Apparently, different disciplines and different applications require different shape descriptions or ‘idealisations’ of reality.

5.3 Perceptive models of assemblies

Different levels of idealization play also a role in the modelling of complex systems. Figure 23 shows a window frame. The frame itself consists of material parts (sills and jambs) which could be modelled as geometric volumes. But in an early design phase, when not all the properties of the constituent parts are known, knowledge may be limited to size and location of the window openings. For this - and for several other applications, such as structural analysis - it is sufficient to describe the openings as topological faces, and the window frame as topological edges which are joined by vertices (Figure 23 b and c).

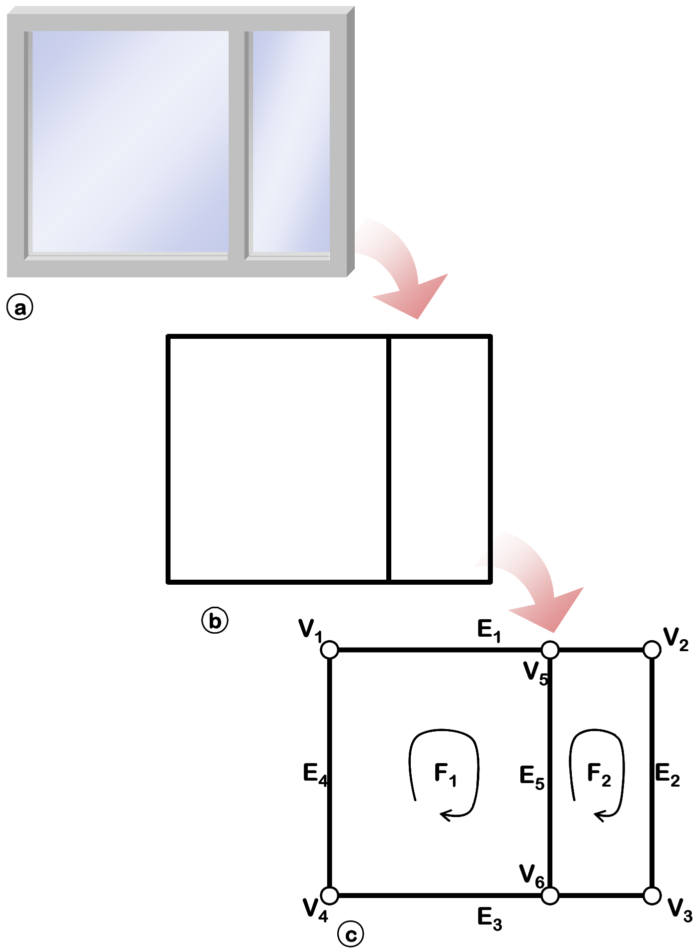


FIG. 23.: Window frame described with Euler topology.

Knowledge about cross-sections of sills and jambs, and details about joints, may be absent in an early phase of design, irrelevant for a discipline, or expressed in implicit form. An example of the latter is that the cross section of a sill or jamb is not expressed in terms of shape, but in terms of section area or moment of inertia. In a design process, it is possible that the moment of inertia is derived from required shape, or that shape is derived from the required moment of inertia.

In the given example, topological structure expresses relative position and connectivity of structural elements. Combined with geometry, it expresses also reference geometry - such as reference points, lines and planes – which plays a significant role in construction. ‘Classic’ Euler topology is not sufficient for this purpose, as will be explained with an example.

Suppose that, in the example shown in Figure 23, the upper sill E1 is bound by vertex V1 and vertex V2, and thus not by vertex V5. Vertex V5 expresses only that one end of edge E5 is located on edge E1. The notion that E1 is not split into two pieces by V5 is important for the calculation of the length of E1. In this case, a different kind of topological association needs to be introduced, namely that of non-bounding enclosure: V5 is topologically enclosed by E1.

An example of this kind of topological association for the window frame is depicted in Figure 24.

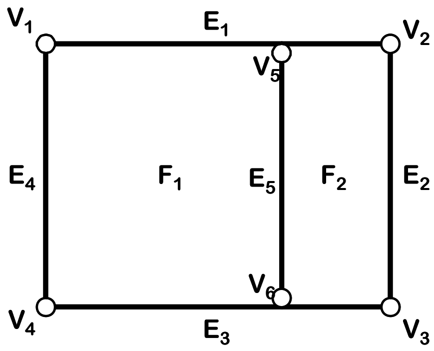


FIG. 24.: Window frame described with Enclosure Topology.

The introduction of Enclosure Topology creates however an issue with the boundary description of faces F1 and F2, because it is not possible to form a closed loop of edges that bound these faces.

This issue can be solved by introducing another feature in this extended topology: that of *edge-side*. An edge-side represents the link between a face and an edge. Face F1 has four edge-sides, including one that links F1 with E1. Edge-side F1-E1 uses V1 and V5 as boundary.

This kind of description may seem to complicate topological structure, but the introduced new entities are all significant for a product model. An Edge-side, for example, may be associated with knowledge that is specific for the interface between the object that corresponds with F1 (a glass panel) and the object that corresponds with E1 (the sill). An example of relevant knowledge is the geometrical offset between the boundary of the glass-panel and the reference line that is associated with E1 (due to the thickness of the sill, the border of the glass-panel does not coincide with the reference line, but will have an off-set of a few millimetres or centimetres).

In the example of the window-frame shown in figure 25, the precise dimensions of a glass-panel are determined by the dimensions of F1 and its bounding edges, corrected by off-set values given by edge-sides F_1E_1, F_1E_3, F_1E_4, and F_1E_5.

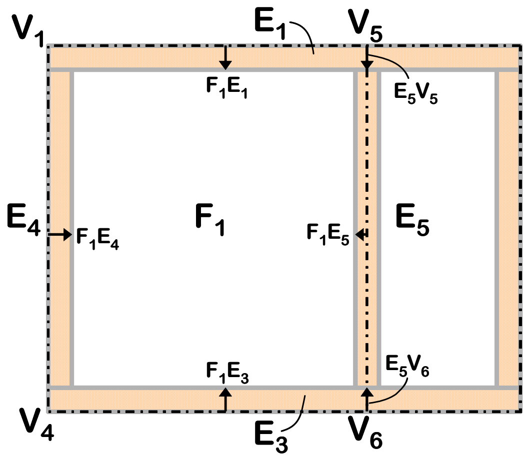


FIG. 25.: Window frame described with Enclosure Topology and Edge-sides F_iE_j and Edge-ends E_iV_j

Following the same line of thought:

- An edge has two edge-ends, each of which links the edge with a bounding vertex. Again, edge-ends are relevant because they can carry relevant information about geometrical off-set and – for example – knowledge about the processing of the ends of a sill or jamb.
- A face had two face-sides, each of which links a face with a volume. Face-sides may carry information about the side of a face that is oriented to a volume (or a space), such as surface-finishing and geometrical offset that results from the thickness of a wall.

5.4 The interpretation of extended topology as connectivity network

Spatial and topological structures can be interpreted as connectivity networks, such as described in sections 2 and 3.

Although some references will be made to the more widely known Euler topology, the present theory makes use of a new kind of topology that is developed by P.Willems [Willems 88, Willems 98]. It will be discussed and applied here in implicit and explicit form.

An introduction to n- and d- topology

The implicit form is independent of spatial dimensions and is therefore named n-dimensional topology (or in short notation *n-topology*), where n stands for any (positive) natural number. This topology is theoretically valid for 0-, 1-, 2-, 3-, 4- and higher dimensional spaces. It can be interpreted as a kind of a connectivity network, to which also the principles of modularity apply.

The explicit form is based on dimensional order, and will be called *d-topology*. Its validity is restricted to 0-, 1-, 2- and 3-dimensional spaces (hence, *d* has values 0, 1, 2 or 3). It is at the same level of abstraction as Euler topology. d-Topology can be expressed as a specialization of n-topology.

Willems' topology (i.e. both n- and d-topology in this theory) offers the possibility to map topological structures on modular networks (m-networks). Or, to be more precise, they are specializations of the connectivity network concept that is described in chapters 2 and 3. This implies that the theoretical principles for modularisation of m-networks applies also to n- and d-topology.

Willems' topology extends Euler topology however also with another notion: the notion of spatial location, which differentiates topological connectivity in boundary topology and enclosure topology.

expressions

notions

		m-network	*n-topology*	*d-topology*	*Euler-topology*
dimensional order		none	implicit	explicit	explicit
spacial location	*boundary*	none	explicit	explicit	explicit
	enclosure	none	explicit	explicit	none
modular network		explicit	explicit	explicit	none

FIG. 26.: A comparison of modular connectivity network, n-, d- and Euler-topology.

Figure 26 shows the abstraction levels of the topologies that will be discussed, as well as the notions that result in their differentiation. Of the expressions mentioned, m-network, n-topology and d-topology are part of the present theory. Euler topology is only shown as a reference.

5.4.1 Euler Topology

Topology, the 'study of position', is concerned with the mathematics of connectivity networks. In its simplest form, a topological network consists of vertices (comparable with nodes) and edges (comparable with links) that connect vertices. A topological network describes what is connected with what. It can, for example, describe how railway tracks connect stations and how a person can travel from one station to another via alternative routes. If projected on a surface, a closed set of edges may enclose a surface, which is called a face in topology. A closed set of faces encloses on its turn a region of 3-dimensional space - a volume. Volumes are thus bound by faces, faces by edges, and edges by vertices.

Topology deals with relative positions, not with absolute positions in a particular co-ordinate system. It doesn't deal either with sizes or distances. These aspects are described with geometry. Yet, topology is not independent of spatial notion: volumes are 3-dimensional spaces, faces are 2-dimensional spaces, edges are 1-dimensional spaces, and vertices are 0-dimensional spaces. Euler topology describes spatial connectivity in specific dimensions.

5.4.2 n-Topology

Intuitively, one would consider vertices in Euler topology as a kind of node in a network, and edges as a special kind of link. But what about faces and volumes? A conceptual extension of networks with concepts that correspond with faces and volumes would make the network sensitive for the notion of dimensions in space.

Willems' n-topology is based on two key concepts, *cell* and *enclosure*. They will be renamed as topological node (NT) and topological link (LT) in the present theory. A topological node (NT) is an abstraction of the concepts vertex, edge, face and volume in classic topology. The latter concepts are considered as NT's of, respectively, dimension 0, 1, 2 and 3. They are symbolized as NT0, NT1, NT2 and NT3 respectively. There are no restrictions on dimensional order, so that the topological rules are assumed to be valid for any dimension n for which n is a natural number.

A topological link (LT) does not exist in explicit form in Euler topology, but is present in many derived theories and practical implementations. An LTn (where $1 \leq n \leq 3$) represents the link between an NT of dimension n and an NT of dimension n-1. If projected on d-topology, LT1 represents the ends of an edge. Every edge has two ends, and multiple ends may be joined in one vertex. LT2 represents a boundary of a face (an NT2) as determined by an edge (an NT1), and LT3 represents a boundary of a volume (an NT3) as determined by a face (an NT2). In section 5.3 an LT1 was called edge-side and an LT2 was called face-side.

In Euler topology, the notion of topological link is referred to as a *manifold*. The kind of topology used to describe volumetric solid objects such as polyhedra, is called 2-manifold, because in this topology every edge joins exactly two faces. If Euler topology is used to describe closed structures of multiple volumes that are bounded by common faces, the number of faces that are joined by an edge may be larger than 2. In fact, the number of faces may be any natural number.

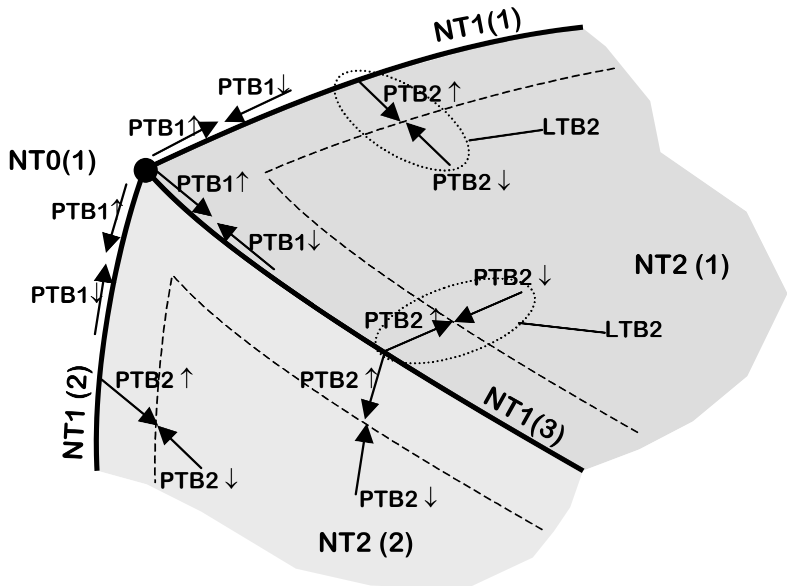


FIG. 27.: Example of boundary topology

An LT requires a further breakdown if it traverses the boundaries of a module. In such a case, an equivalent for the port concept in the connectivity network theory must be included. It is possible to decompose all links into pairs of ports, so that a link can be redefined as a pair of connected ports in the form of a two-tuple: LT(x) = PT(y,z), where x, y an z represent instances of LT and NT.

As LT's are links between NT's of different dimensionality, the two ports that form an LT are asymmetrical. Ports in n-topology will be differentiated into ports of higher dimensionality PT↑ and ports of lower dimensionality PT↓. If dimensional order is made explicit, this notation can be extended with the dimensional notion *n*, thus extending the notation as PT*n*↑ or PT*n*↓. For example, a PT2↑ is a port of a face (i.e. a topological 2-space) that can be linked with a PT3↓, which is a port of a bounded volume (i.e. a topological 3-space).

Figure 27 gives an example of a boundary topology where two faces, NT2(1) and NT2(2), are joined by edge NT1(3), and three edges NT1(1), BT1(2) and NT1(3), are joined by vertex NT0(1). This example shows also topological ports PT's - more specifically topological boundary ports: PTB's. Two ports, such as a PTB2↑ and a PTB2↓, may form together a topological link, such as a LTB2.

n-Topology plays a role in the present theory because it forms a conceptual bridge between the general principles of networks, as discussed in chapters 2 and 3, and topologies with explicit dimensions, as discussed in the present chapter. Furthermore, the concepts of topological link and topological port are relevant for the modular description of product structures, especially if connectivity networks cross the borders of knowledge modules.

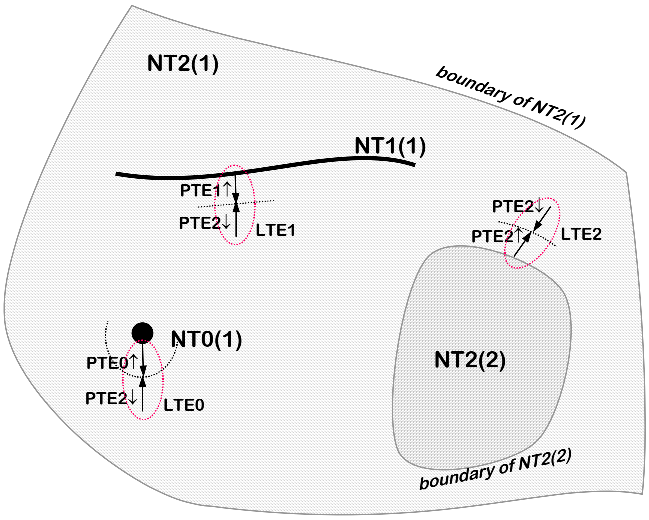


FIG. 28.: Example of enclosure topology

Figure 28 gives an example of enclosure topology where face NT2(1) encloses: a face NT2(2), an edge NT1(1) and a vertex NT0(1). This example shows also topological enclosure ports (PTE's). Two ports, such as a PTE2↑ and a PTE2↓, may form together a topological link, such as a LTE2.

5.4.3 d-Topology

d-Topology is principally the same as n-topology, but it is restricted to 0, 1, 2 and 3-dimensional topological nodes and makes dimensional order explicit. d-Topology contains therefore a large number of concepts that may make implementations complex and difficult. It is only used here as an intermediate representation between Euler-topology and n-topology, and for reasons of clarification and exemplification.

5.4.4 Spatial Enclosure

The links between topological nodes discussed up to this point connect nodes of dimension n with nodes of dimension n-1. The latter nodes bound the nodes of higher dimensionality. In d-topology, for example, an edge is bound by one or two vertices, and a face is bound by one or more edges.

n-Topology as described by Willems offers however also the possibility to define non-bounding links between topological nodes. In that case a node of dimension n may be linked with a node of dimension a, where ($a \geq n$). Node NTn must then be enclosed by node NTa, which means that the first is not located outside the boundary of the latter.

If projected on d-topology, this implies that a vertex may be enclosed by another vertex, an edge, a face or a volume. Or, in other terms, a vertex V may be located on another vertex, edge, face or volume, inside (or on) its boundaries.

5.4.5 Topological Orientation

All topological links that are discussed so far are unilateral. A difference between boundary topology and enclosure topology is however that they have opposite orientations. In boundary d-topology, an NT of dimension n is defined by reference to NT's of dimension n-1. This implies that the existence of the first depends on the existence the latter. For example, an edge in Euler topology is defined by reference to two vertices. Without the vertices, the edge cannot be defined.

The NT that plays the role of boundary does not require a reverse reference, because its existence does not depend on the NT of higher dimension. For example, a vertex in Euler topology may be defined without restrictions on the (number of) edges that use it as a boundary; it may exist even without an edge that refers to it.

In enclosure topology the situation is opposite. The enclosed NT must refer to an NT of the same or higher dimension because the validity of this topological link depends on existence of the latter. The enclosing NT does not depend on existence of the enclosed NT.

6. DISCUSSION OF SOME KEY PRINCIPLES

Up to this point, the latest version of the theory has been described. In this chapter a bit more background information is given about key principles. It is also motivated why some principles were changed compared with previous versions of GARM.

6.1 The Hamburger Model

Probably the most cited and used part of the original GARM is the 'Hamburger' concept. This part of the theory will be discussed here in more detail.

Composition structures are often modelled as hierarchical structures, or 'tree-structures', such as depicted in figure 1. Also buildings can be modelled like this: the building itself is considered as the 'whole', and building elements, such as walls, floors and roofs, are considered as 'parts'. Each element can, by itself, again be seen as a 'whole' that contains 'parts'. The objects in such a tree-structure can thus play two roles: that of 'whole' or that of 'part'. This notion becomes relevant when the information ownership of complex products such as buildings is considered. In the design and specification process, it is usually the architect who is intellectual owner of the 'whole' of the building. The architectural design process stops at a point where components, such as building elements, can be procured.

For the construction, maintenance and operation of buildings it may be necessary to have more detailed information about sub-systems and components of a building. This information is not created by the architect but by other disciplines, such as consultants or suppliers. The 'parts' in the architectural point of view are 'wholes' for these disciplines. This observation can be graphically depicted by splitting each sphere, denoting an object (such as shown in figure 1) into an upper halve and a lower halve. The 'hamburger' simply expresses the idea that objects in a building can have two intellectual owners: the one responsible for the whole and the one responsible for the part. This notion is especially important for construction, as it marks also the most relevant transaction points between disciplines and organizations in a supply chain.

From a transaction point of view, it can be stated that the discipline that is responsible for a 'whole' defines requirements for the 'parts' so that a specialist or supplier can offer a solution for it. From a design process point of view, it can be stated that the designer tries to solve a design problem by decomposing it into smaller problems, up to a point where available solutions can be found. If the latter point of view is taken, it means that the use of 'Hamburgers' does not have to be restricted to transaction points, but more generally as a means to support a problem solving process.

Consequently there are, for each design object, two types of specifications: one that defines requirements for it, or considers it as a design problem, and a second that describes the solution.

During a design process it may occur that only the problem is known and/or that requirements are specified, but that no solutions are (yet) available. Subsequently, the search for a solution may bring up several alternative solutions from which a designer has to choose. At the end, one of them has to be selected. This process is depicted in figure 29.

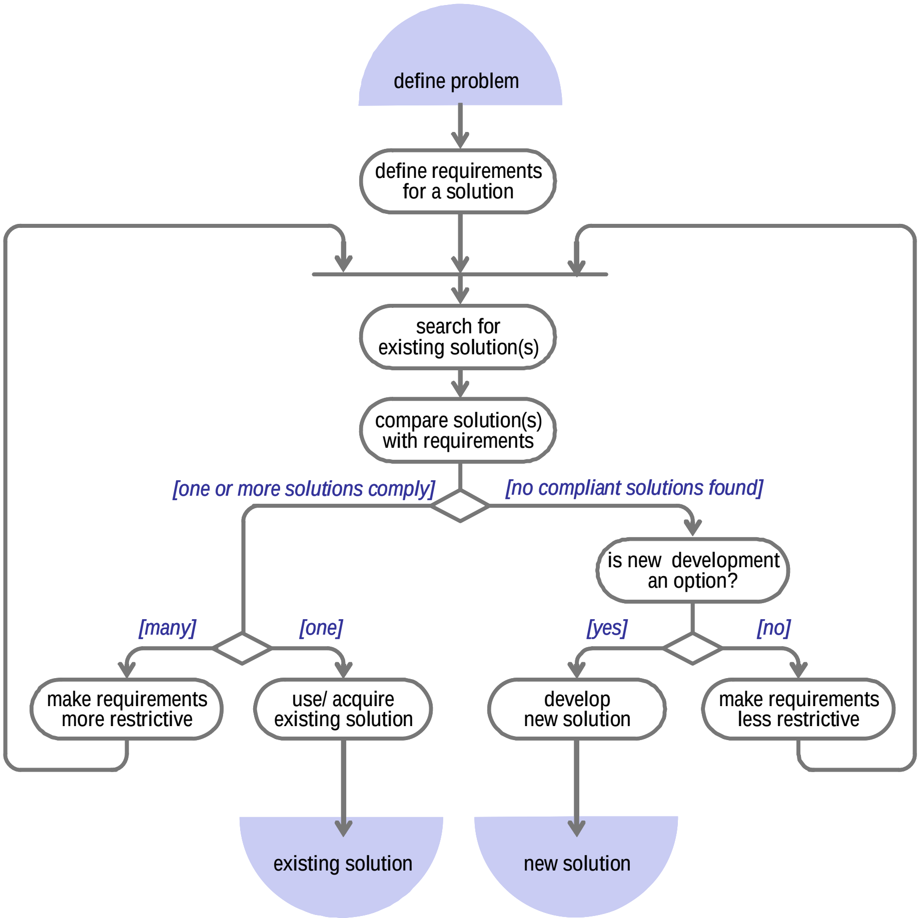


FIG. 29.: The Hamburger concept as applied in a problem solving process.

In a dynamic design process it may occur that requirements change so that a once chosen solution has to be departed. If both problem and solution are described in a well formalised way, it is possible to check automatically whether selected solutions remain valid or not. By keeping 'problem' and 'solution' separated, it becomes also possible to reuse solutions.

The Hamburger concept is often explained in the context of top-down design processes. But it does not prescribe any design method. It can also be applied in bottom-up or omni-directional approaches.

The first version of the GARM used the term 'Functional Unit' for the upper part and 'Technical Solution' for the lower part of the Hamburger. These terms *combine* the transaction way of thinking with the problem solving way of thinking. It would be more pure to call them 'problem' and 'solution', or 'required product' and 'offered product'. In a design process, the distinction between these two meanings seems to be less important: if an existing product is not capable to meet the requirements of the designer, it becomes automatically a design problem. Therefore these two interpretations of the Hamburger concept were retained as part of the Specification Hierarchy. Also, many people were already used to the original terms.

In a more generic sense, the Hamburger model supports the modelling of modularized hierarchical structures. As such, it can be detached completely from transaction processes or problem solving processes. For that reason the concept is presented in this paper first as the generic 'Slot'/'Module' concept.

In all product/process modelling oriented research projects done so far, it appeared to be sufficient to have only two types of modular hierarchies, namely Specification Hierarchies and Installation Hierarchies. For these, the terms Functional Unit/Technical Solution and Functional Item/Physical Item are proposed.

6.2 Modularization of complex parametric models

As stated, an important property of the Hamburger concept is that it makes composition hierarchies modular. This appears to have important advantages.

In a project for the modelling of a viaduct family, which is described in more detail in section 7.2, it was first attempted to develop a single parametric model. This appeared to be impossible, due to the huge number of variables and rules needed to make the model semantically consistent. Subsequently, the model was hierarchically decomposed into a Specification Hierarchy, in line with the principles of the present theory. This breakdown of the model into many small generic modules (Technical Solutions) reduced the complexity of the model drastically. An implication of modularization is that parameters are defined locally in Technical Solutions. Global parameter-values are defined in top-level Technical Solutions, and may then be transferred to lower level modules via Functional Units. In this process, parameters may be converted. For example, in a model of a viaduct, the highest level Technical Solution may contain a parameter for the road width. This road width influences the dimension of various components, such as the bridge deck and supporting pillars. However, for a parametric model of a pillar only the width of the bridge deck may be relevant, not the width of the road. In the top-level model of the viaduct, road width is converted into the width of the bridge deck so that this parameter value is subsequently transferred to the pillar model.

It is also possible to transfer parameters between modules on the same level via Ports. In such cases they express boundary conditions for the application of a module.

The modularization principles appear also important for the configuration management of systems that change, such as caused by maintenance or renovation, for the expression of design variants of almost similar systems - such as described in GLT case, section 7.4 - and for the 'plug and play' use of predefined solutions in a solution knowledge base.

6.3 Building performance modelling

While in traditional building projects clients prescribe a building in full technical detail, there is a trend in construction to move in the direction of performance based specifications and contracts that prescribe only 'why' and 'what' but not 'how'. It is then up to the supplier to propose a solution.

This principle is supported by the Hamburger model, in particular in the transaction between client and supplier, and between supplier and sub-suppliers. Functional Units specify client/user requirements, and Technical Solutions specify the offerings of suppliers. This concept was used in research on performance based contracting (Foliente 2005, Szigeti and Davis 2005), building specification modelling (van Rees 2007), product selection from e-catalogues (Jain and Augenbroe 2003) and emerging standards for electronic tendering (Woestenenk 2004).

According to the theory presented here, usage of the Hamburger principle should be consistently used within Specification and Installation hierarchies, not just at transaction points between organisations. The theory does not prescribe 'what', 'why' and/or 'how' has to be built. Also details about requirements and performances are not addressed here, as these may vary from application to application. The theory simply provides a framework that can be filled in in many different ways. The above cited references describe valuable applications of this principle.

6.4 From a framework of orthogonal dimensions to notion-theory

The IMPPACT and PISA reference models (Gielingh and Suhm 1993, Gielingh et al 1995) presented the main principles in the form of a multi-dimensional framework. The IMPPACT model recognized, for example:

a. Generic, Specific, Occurrence
b. Required, Proposed, Realized
c. Static, Dynamic
d. Design, Production, Demolition (changed by PISA into Construction, Operation, Demolition)
e. Shape, Material

It is possible to use the ordinates in each dimension as adjectives for the noun *Definition* or *Product Definition*. As the dimensions are orthogonal, ordinates from different dimensions can be combined. For instance, we can talk about a 'Specific Required Shape Definition', or even an 'Occurrence Realized Production Material Definition'.

The idea of orthogonal dimensions appeared to be powerful, enabling the definition of very precise semantics. Modelling methodologies, however, do not accommodate this principle; they force model developers to define the classes that result from this multi-dimensional framework. If all dimensions are combined, this results in 3*3*2*3*2 = 108 different classes. If not all dimensions are combined, so as to define also generic classes, this results in several hundreds more classes. These class definitions are quite inefficient and highly redundant, given the fact that the 'master-model' is so simple.

In a more recent study (Gielingh 2005, Gielingh 2008b) the cause of this friction between GARM dimensions and modelling methods is found. In the Theory of Definition, developed by Plato and used by Aristotle for his categories (Barnes 1984), classes result from an abstract super class, called Genus, and one or more differentia. The set of differentia that is required to define a concept is also called the intentional definition of this concept. The GARM dimensions are indeed differentia according to Aristotle's theory.

In the referred study, it is concluded that differentia actually depend on the existence of perceiving systems. A perceiving system, such as an actor, uses a perceptive frame, which consists of one or more notions. A notion is the ability of a perceiving system to observe differences between phenomena. Hence, the perceptive frame determines how a Domain of Discourse is classified. Different observers of a Domain of Discourse will use different perceptive frames which consequently result in different classifications. This is the reason why it is so difficult, if not impossible, to reach consensus in standardization efforts about conceptual models. A first step towards resolving this problem is to recognize the existence of different perceptive frames. Perceptive frames can be modelled by means of notions. An example of such a notion is the notion of change: does the observer recognize change in two successive observations of a phenomenon? If so, the observer will classify the phenomenon as dynamic (i.e. a process). If not, the observer will classify the phenomenon as static. It is impossible to determine in an absolute sense whether phenomena are static or dynamic: in an infinite short moment of time almost everything will look static, while in an extremely long period of time almost everything seems to be dynamic. Like beauty, also classes are 'in the eye of the beholder'!

It is possible to redefine the GARM principles using the Notion-theory. But as this theory is still subject of research, the present article does not yet use Notion-theory as a means for formalization.

6.5 On the lifecycle dimension

Early versions of the GARM contained a lifecycle dimension in the form of 'as designed', 'as built', 'as operated', 'as maintained' and 'as demolished' product specifications.

This appeared to be insufficient for the description of a building lifecycle. There is not a single 'as designed' or 'as built' specification; there can be many different ones. There is however a fundamental difference between imaginary product specifications and product descriptions that result from measurements of the actual product. This notion resulted in the 'imaginary/actual' principle. Conceptually, there is no fundamental difference between 'as built', 'as maintained', 'as operated' and 'as demolished' specifications. The activity of maintenance may consist of partial disassembly and reassembly of a product. Such activities will now be modelled as processes, using the static and dynamic views. Design processes, building processes and user processes can be described in the same way; at the abstraction level of the theory there is no need to differentiate between them.

If there is a difference between an operational view and a building/maintenance view, caused by different perceptions of actors, then these will be considered as different discipline views of the product model (section 2.4). This new way of lifecycle modelling is applied and verified in two projects - IMPPACT (1989-1992) and GLT (1996-1999).

6.6 On the specification dimension

In earlier publications, a distinction was made between generic, specific and occurrence information. The generic/specific principle has never been questioned and appeared to be useful in all projects. But the IMPPACT project revealed that the occurrence, which was defined as an individual object, was not sufficient. There was also a need to describe the various stages of an individual object when it goes through subsequent phases of its lifecycle.

The lifecycle oriented GLT project proved this need once more. Therefore it was decided to introduce the concept 'individual', which replaces the former 'occurrence', and to reserve the concept 'occurrence' for a product description for an infinite short moment in time. Hence, a single individual object exists between its physical birth and physical death. During this life it will undergo many changes. These changes are processes. An individual object has many occurrences: one occurrence (static description) for each relevant phase of its life. Between occurrences are processes that change the object (dynamic description).

The older specifications were not entirely clear about the scope of generic and/or specific product descriptions. A 'specific product description' contained all information of an object, except location in time and space. This rule is now dropped. Generic or specific product information may cover aspects or properties of a product in stead of information about the whole product. For example, generic or specific descriptions may be given for shape and material separately. Only at the occurrence level, these properties are combined. This means that shape, material or other information may be shared by multiple occurrences. In figure 14 an example is given of a vase where occurrences 4.1, 4.2, 4.3 and 4.4 share the same shape information, while occurrences 1.1, 2.1, 2.2, 3.1, 3.2 and 4.1 share the same material information. This is because in the first stage of production of the vase shape is modified, while in the last stage material properties are modified. This approach reduces information redundancy.

6.7 On views

Older publications of the theory were not specific about the impact of different views on a system. Different views may result in different networks and even in different compositional hierarchies. To allow different hierarchies would however destroy the simplicity of use, in particular in the selection of Technical Solutions for Functional Units. It was therefore decided to allow only one specification hierarchy.

This decision may seem restrictive but has important advantages, also for the design process. Each module (i.e. a Technical Solution or Technical Individual) has an intellectual owner. This 'owner' is a person, discipline or organization that has final responsibility for the solution or the individual. This owner determines the next level of decomposition by the selection of lower level modules. Within such a hierarchy there may be multiple intellectual owners, but only one for each module. Consequently, there will be only one master hierarchy. The owner of a module has the primary view. All other views are secondary. It is then possible to represent different views within a module by means of different networks, such as described in section 2.4.

It is possible to derive different view-dependent hierarchies from a Specification Hierarchy, but these are copies while the latter remains the 'master'.

This approach has the benefit that product specifications become highly modular, less messy and easier to manage. Modules can be archived for future reuse. As this principle influences the way of working in design, the theory can become part of a methodology.

6.8 On installation hierarchies

Older versions did not contain the principle of Installation Hierarchies. The need for Installation Hierarchies was found in the GLT project, which required the modelling of the full lifecycle of 29 almost identical plants. Maintenance and Operation require information about individual objects. For the modelling of the object lifecycle it is essential that data are collected for each lifecycle phase. The latter is addressed by occurrences in the present theory. Also the separation between functional and physical items appeared to be very important, such as demonstrated by the example in figure 15. The dichotomy between the functional view of an operator and the technical view of maintenance personnel is addressed by the Installation Hierarchy. Information that is the same for each of these individuals should be placed in the Specification Hierarchy, so that data redundancy will be reduced.

6.9 Is it a model or a meta-model?

During the development if ISO 10303 it was noticed that the presented theory, which was referred to as the General AEC Reference Model, is at a different kind of abstraction level as most other models being discussed. It was actually seen as a planning model for STEP by some (Kemmerer 1999), or as a meta-model by others.

This issue became very clear during the European research project IMPPACT. The present theory formed the basis for the IMPPACT Information model that was used for the sharing of information between dozens of CA-applications. Within this project, there was also a task for the parametric description of features, for which the PDGL language was developed (Bjørke, Myklebust, 1992). It appeared that some of the concepts defined in the Reference Model, in particular the Generic/Specific/Individual/Occurrence dimension, are in more simplified form also part of the meta-model of Express. In this language, like in many other languages, a distinction is made between entity-types and entity instances. PDGL used the same principle by using types for the parametric description of features and instances for the individuals. This implied that all principles for the Generic/Specific/Individual/Occurrence dimension could not be used in conjunction with PDGL. Languages such as Express apply also hierarchical structures, such as for generalization hierarchies, and apply network structures via references to other entity types. The issue was described in (Gielingh and Suhm 1993).

This issue was further investigated in project PISA (Gielingh et al 1995). An attempt was made to incorporate the GARM principles in a meta-model, resulting in a graphical 'GARM language'. Although this 'language' appeared to be powerful, it appeared not to be suited for practical use due to its richness. For end-users and model developers simple languages are needed, of which the meta model contains just a handful of basic concepts.

IMPPACT reduced the GARM principles into a model of five orthogonal principles, also called 'dimensions'. As these principles can be combined, they result in a wide variety of specific concepts. In total more than 500 different concepts (or entity-types) could be defined with these five IMPPACT dimensions. For this reason, most implementations of the theory used only a limited number of principles and resulting concepts.

The Theory of Notions that was already mentioned in 6.4 (Gielingh 2008a) removes the artificial distinction between models and meta-models, solves the abovementioned problem, and simplifies the modelling of conceptual structures.

Despite these shortcomings of current modelling languages, it has been possible to implement and apply (parts of) the theory in practice, as will be discussed in the next chapter.

7. APPLICATIONS

The concepts that are presented here have a long history of application, verification and critique. The theory evolved over time. Each principle presented here has been implemented and verified at least once. Four important implementations will be discussed here.

7.1 Design and work preparation of building wall systems (1982)

In 1982 a parametric design system was developed for Interfinish, a manufacturer of flexible interior walls for office buildings. It formed the basis for the current theory. At this time there was no software available that could be used as a basis; the author had to write all software, including drivers for the vector-display and the A0 plotter.

An example will be given for the design of a small office building. This example comprises the complete process from initial requirements specification to the specification of the tiniest parts, such as steel profiles, drilled holes, screws and paint. For reasons of conciseness, only a thin 'slice' of the specification will be discussed.

Figure 30 gives an example of five networks, each providing a different view on the office building. On top, a model is shown of the organization that will be housed in the office building. The second model shows the staffing of this organization. This second model is needed because certain functions of the organization may be combined by a single person, while other functions are executed by multiple employees. The third model describes the activities that need to be supported by the office building. The fourth model translates these activities into a special model: certain activities can be combined and supported by a single space.

Activities that occur concurrently need to be separated from each other, in order to avoid disturbance. However, there may also be a need for interconnection. This small example shows only two types of connection: mechanical connection (or: access via a door), and visual connection (or: the possibility to see one space from the other space, which can be facilitated by a window). Other types of connectivity are, for example, daylight access and acoustical access. The isolation of one space from another can also be constrained. For example, if one activity produces noise while another activity requires quietness, this can be translated in the design by placing these activities far apart, or by erecting a wall between them that has high acoustical isolation.

The fifth model, finally translates the spatial model into a model of rooms, separated by floors and walls, as well the connections between these building elements. This is a topological model of the office building.

The theory does not prescribe which aspects of a building are being modelled; it supports any kind of model, provided that it can be interpreted as a network. Chapter discussed how a topological model can be interpreted as a network.

The five models form different but interconnected networks, compliant with the principles described in chapter 2.4 (multi-view networks). They support the process of functional specification, by translating organizational needs into requirements for the building, its spaces and its main construction elements. They enable also the tracking and tracing of design history and design changes. For example, the model makes clear that space S2 is a meeting room that can be used for two different activities: client meetings and work meetings.

Figure 31 presents a simplified composition view of the Specification Hierarchy, in line with the principles described in 4.5. The five networks presented in figure 2 are part of module A.1 on the top. But for reasons of simplification only the two last networks are shown here, with only four nodes in the network(s): the meeting room, the circulation space and two walls.

Wall 1.1 separates the meeting room from the circulation space. This means that these two spaces pre-condition the wall: characteristics of the spaces and the activities that take place within them entail requirements for the wall.

Wall 1.1 is defined here as a Functional Unit. A solution for this Functional Unit will be created as Technical Solution Wall B.1. It is possible that this same Technical Solution meets the requirements of other Functional Units so that it can be reused.

Wall B.1 has a structure that will be described in more detail hereafter. Only two parts of this structure, a jamb and the moving part of a door, is shown here, The Functional Unit 'door' is fulfilled by Technical Solution E501.

The door is connected with the jamb with two hinges. These hinges are considered to be part of door E501. They are connected with the rest of the door and jamb via bolts that fit precisely in through holes in each hinge. There are four of these needed for each hinge.

In the decomposition structure shown in figure 30, the link between door and doorframe is decomposed using ports. The connection of each screw (module D.1) with each screw hole (D.2) can be traced via the higher levels C.1-C.2 and finally via B.1. A connectivity view of this part of the model is sketched in figure 41.

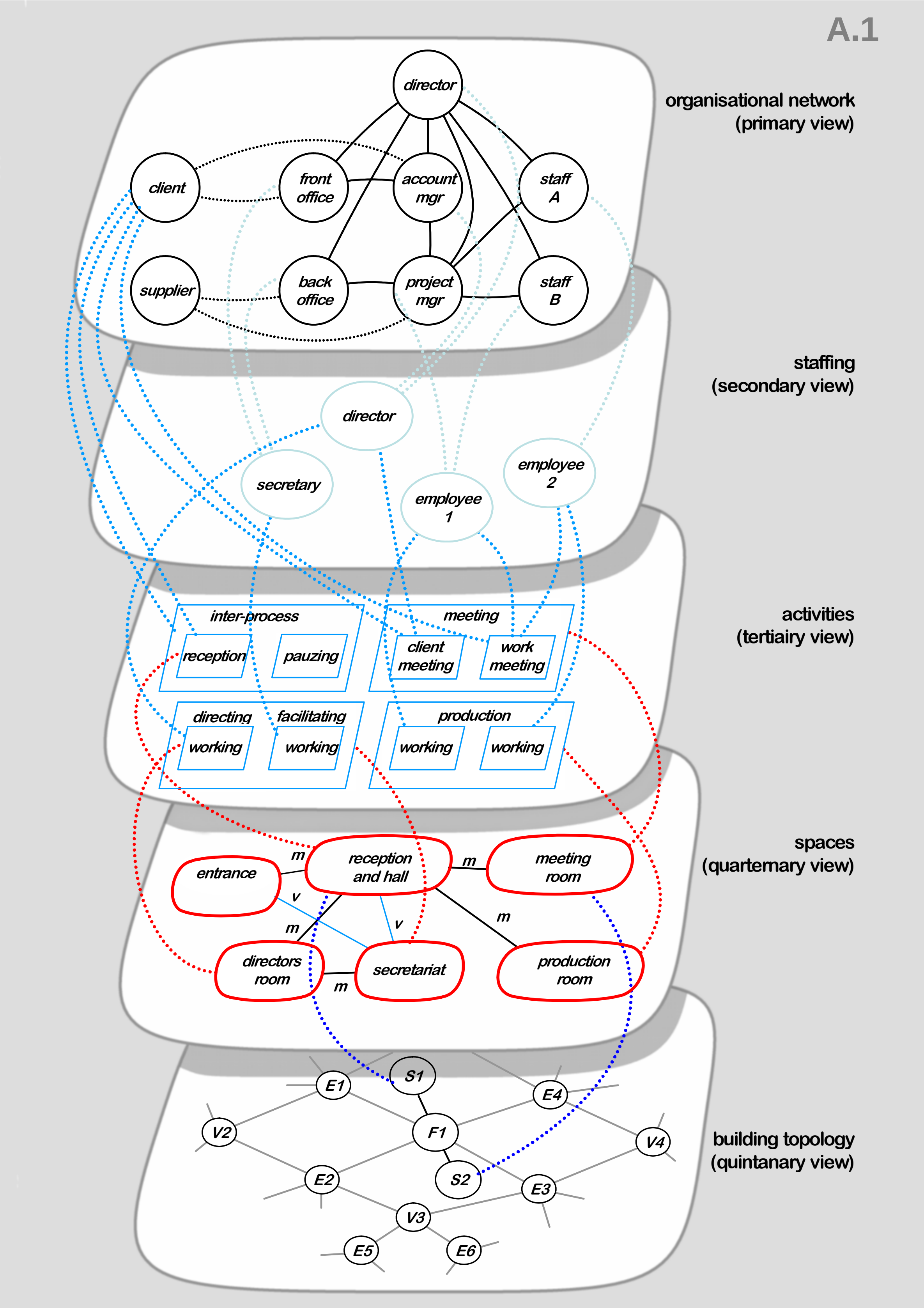


FIG. 30.: Five different views on the functional network of the office building.

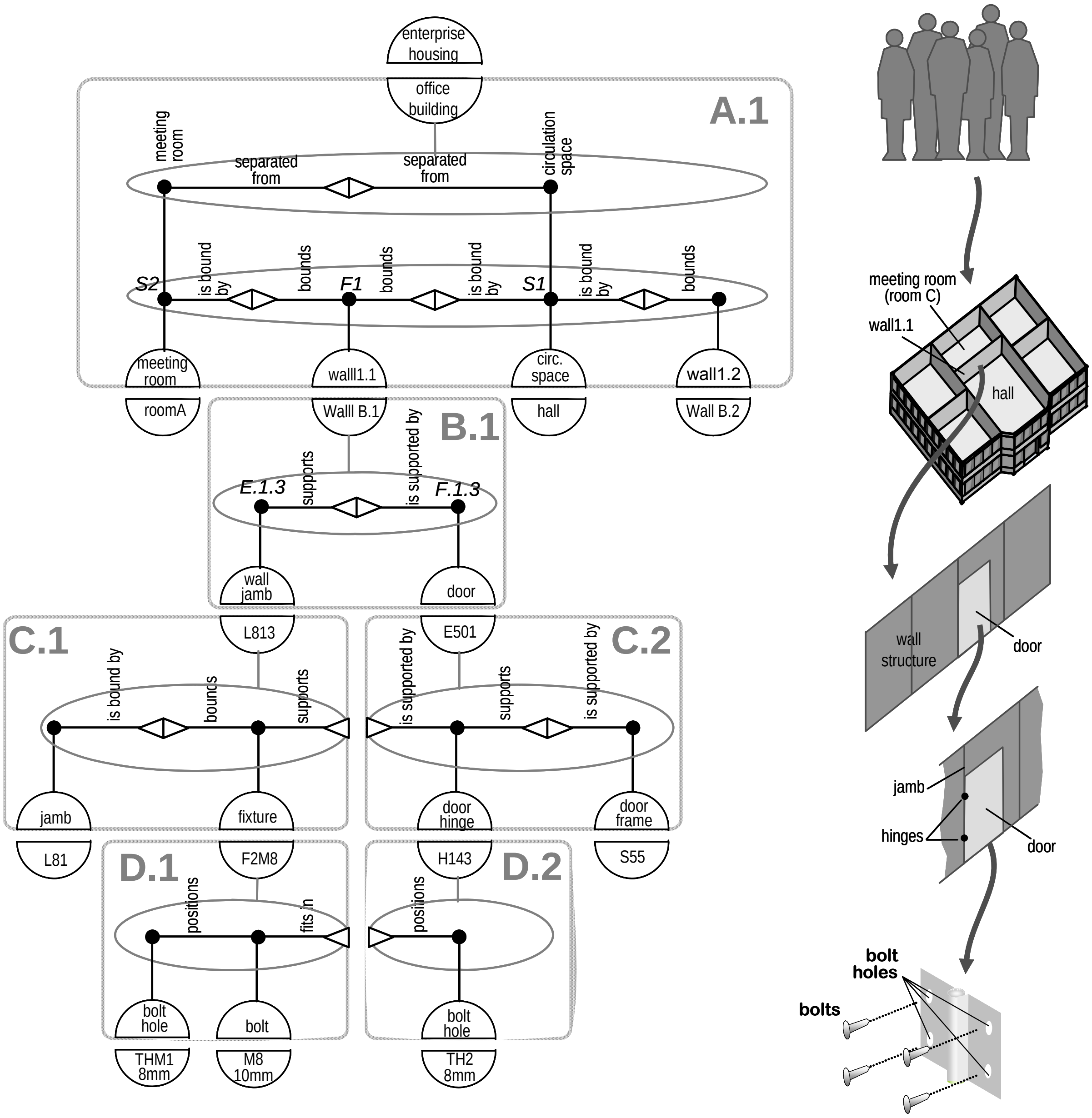


FIG. 31.: Simplified example of the Specification Hierarchy of a small office building.

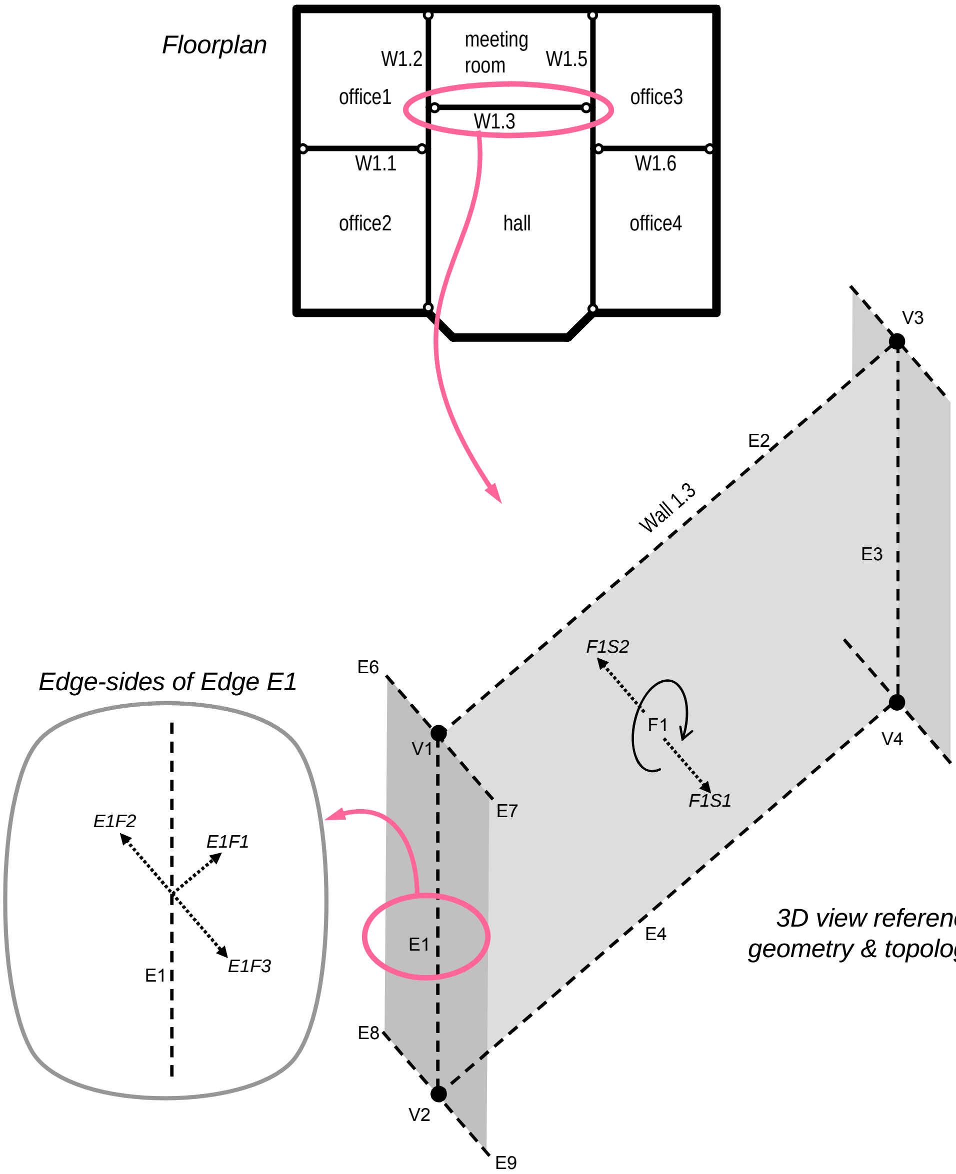


FIG. 32.: The floor plan (top) is topologically decomposed into reference geometry; walls are denoted by lines, line connections by small circles. A 3-dimensional view of Wall 1.3 is shown below.

The top of figure 32 shows a 2-dimensional floor-plan of the office building. It is topologically described, where walls are depicted as lines and connections between walls as small circles.

Wall W1.3, which separates the hall from the meeting room, is shown in 3-dimensional space below. The wall is topologically represented by a face F1. It has two face-sides: F1S1 is oriented to space S1, the hall, and F1S2 is oriented to space S2, the meeting room. Boundary conditions or requirements that a space has for the wall are transferred to the wall via these face-sides. For example, it may be decided that the hall is colored grey: this forms a condition for face-side F1S1. The other side of the wall, represented by F1S2, may be colored white. Also characteristics of the spaces or the activities that take place in them, such as temperature or noise-production, which may condition the selection of technical solutions for the wall, are accessed via the face-sides.

Face F1 is topologically bound by four edges. Of these, edge E1 connects face F1 with faces F2 and F3. Technical Solutions that will be chosen for the connection represented by edge E1 may provide boundary conditions to wall W1.3. The solution shown in figure 33 requires an off-set value for the dimension of Wall W1.3 with 35 mm. In other words: the wall does not extend from E1 to E3, but has to shortened a bit because of the material thickness of the wall to which it is connected. This off-set value is associated with Edge-side E1F1.

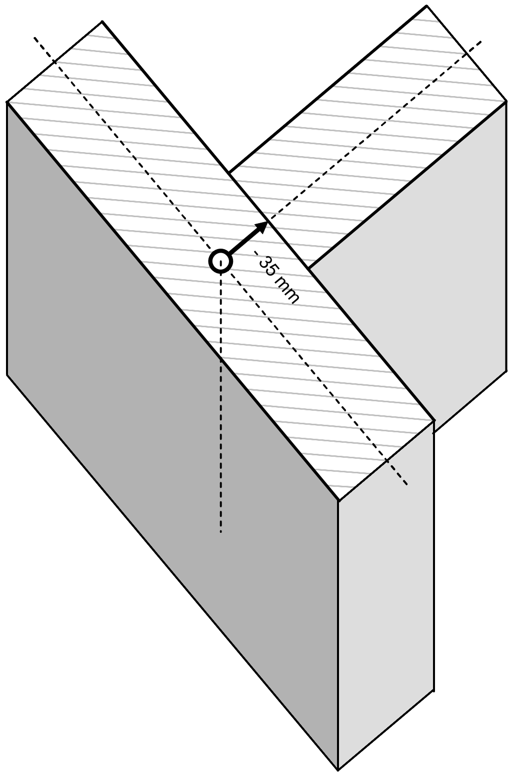


FIG. 33.: Edge-side E1F1 contains knowledge about the off-set value of wall F1.

Figure 34 shows that the distance between E1 and E3 is 4800 mm. After subtraction of the off-set values, the calculated net-width of wall W1.3 is 4730 mm.

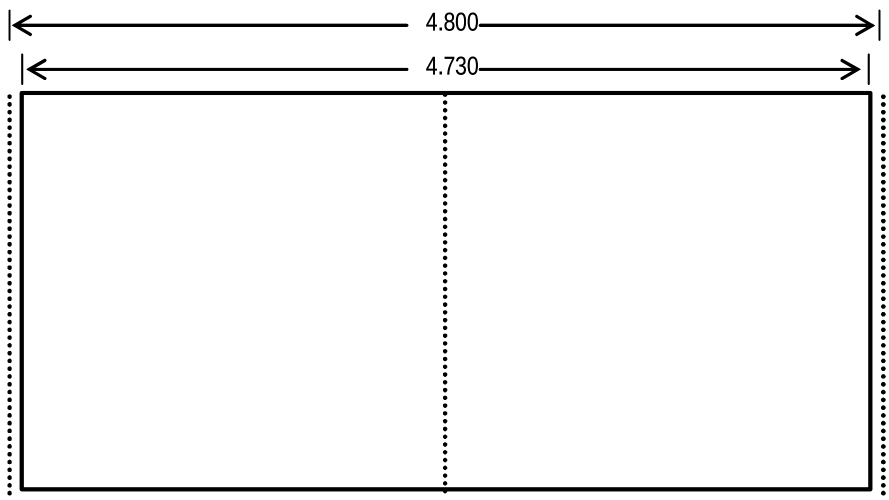


FIG. 34.: The off-set value is subtracted from the nominal dimension of wall F1 (4800 mm) in reference geometry to obtain the required nominal dimension in material geometry (4730 mm).

In a next step, the wall is partitioned into segments, each corresponding with a module, see figure 35. The maximum segment-width is 1200 mm.

It is then possible to fill the segments with modules. A library of standard, parametrically described modules is available. Figure 36 shows that modules can be placed in segments via a drag and drop user-interface.

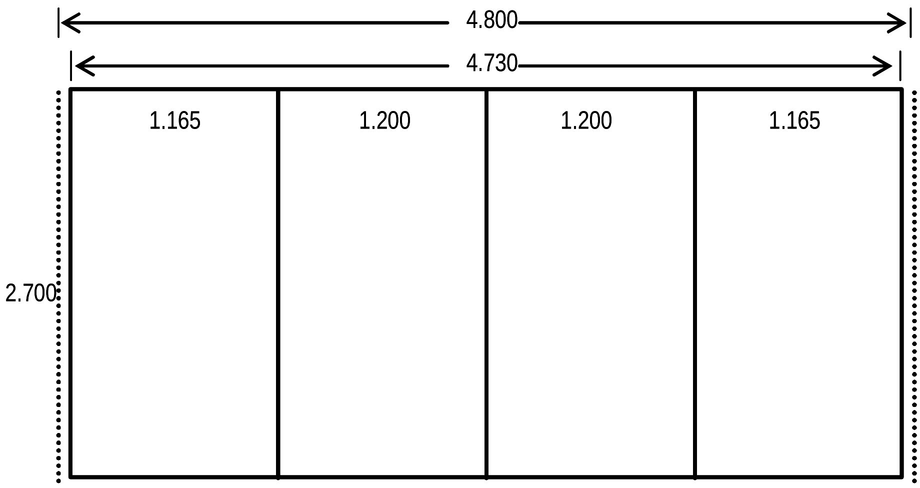


FIG. 35.: The wall is now partitioned into segments for wall modules (maximally 1200 mm width).

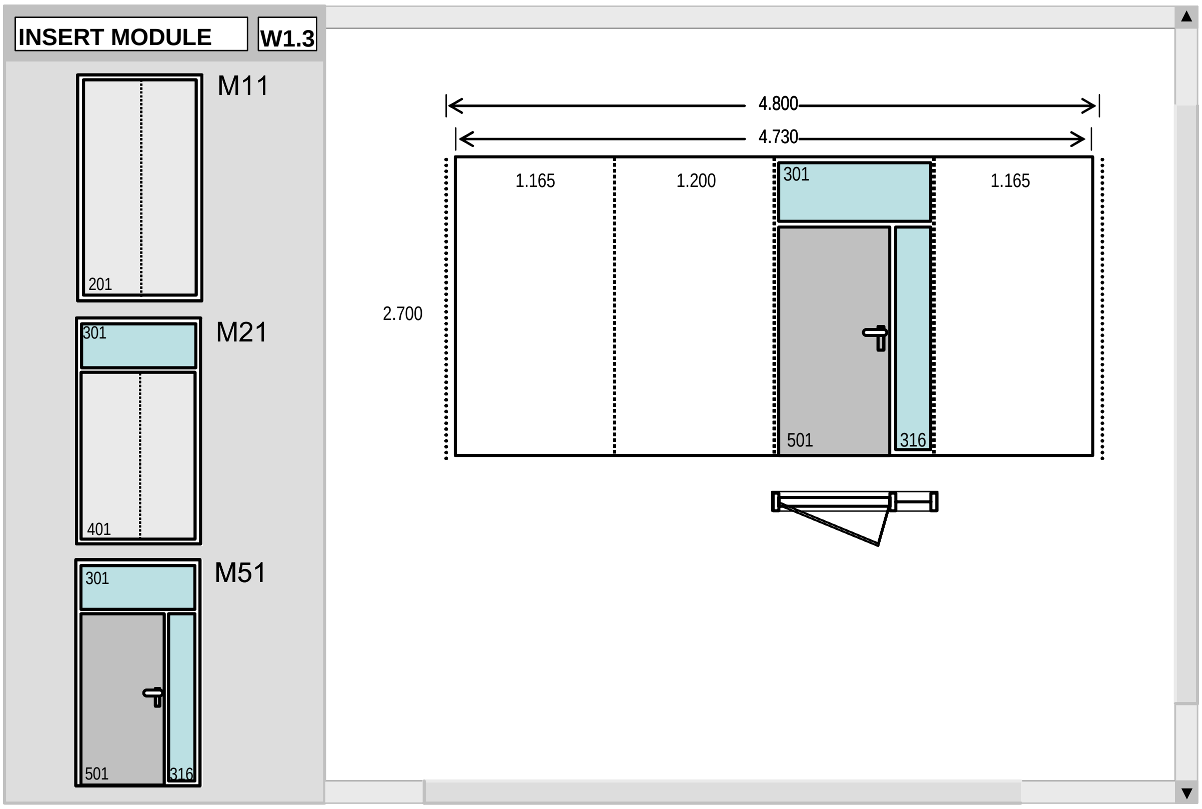


FIG. 36.: For each wall module location, a selection is made from predefined wall modules in the module library, based on a 'drag and drop' user interface. These modules are parametric. Most parameter values are derived automatically from functional requirements. Size of the module is determined by the dimensions of the module frame.

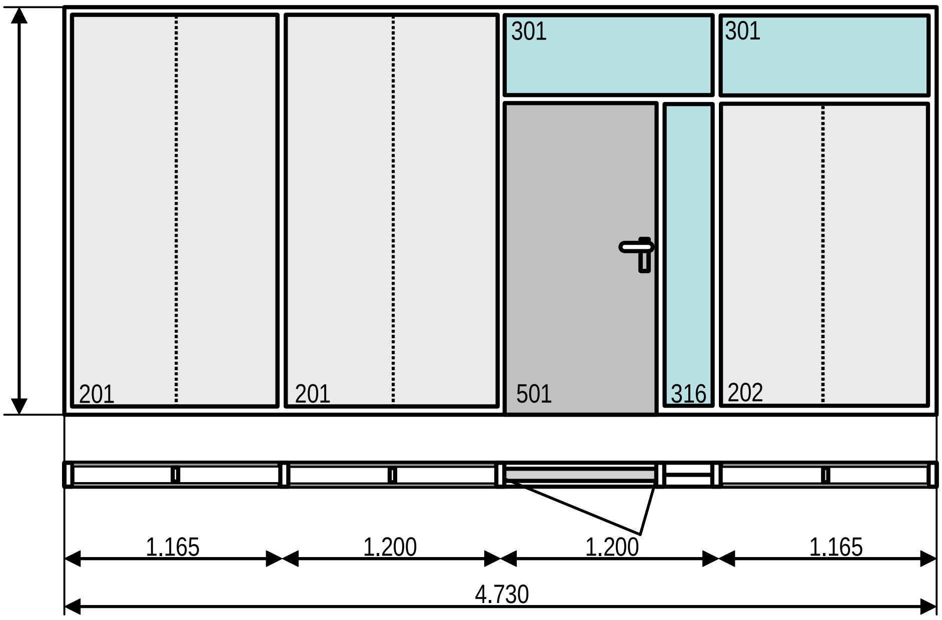


FIG. 37.: Once the wall is defined, the application can produce drawings automatically..

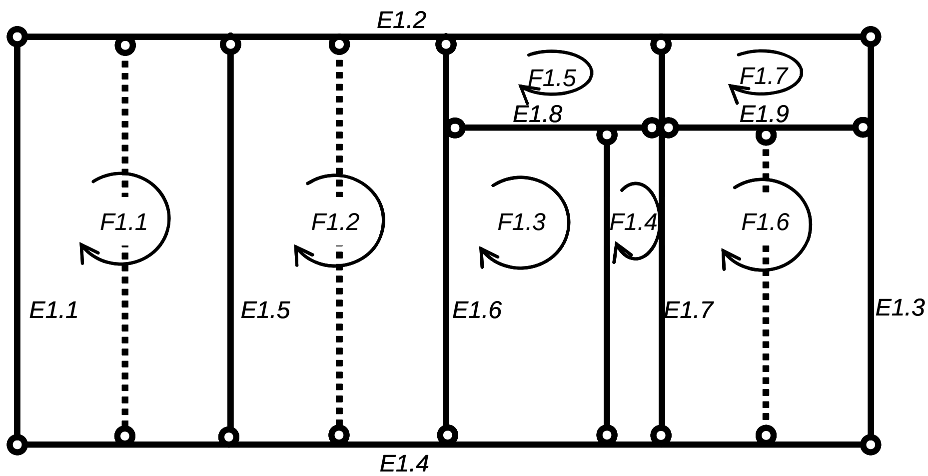


FIG. 38.: With the wall modules in the product library, the topological structure of the wall frame is automatically defined. With this knowledge, and with knowledge associated with linear elements (edge-sides) and nodes (edge-ends), the actual dimensions of steel profiles, panels and the location of form features are calculated. Dashed lines show non-bounding edges (i.e. edges on a face). They represent jambs inside the wall frame. Hence, Face F1.1 is bound by Edges E1.1, E1.2, E1.4 and E1.5.

Once all modules are placed – see figure 37 - the application calculates the resulting topology of sills, jambs, elements and connection nodes – see figure 38. This topology forms a network in which n-dimensional objects condition (n-1)-dimensional objects and reversely. More concrete: the 2-dimensional elements, such as wall panels, doors or windows, provide requirements for solutions for the 1-dimensional jambs and sills, while the jambs and sills provide requirements for solutions for the 0-dimensional nodes. Then, after technical solutions for the nodes are chosen, the nodes return boundary conditions to jambs and sills, such as required off-set values and/or modifications needed for making the connections, and the jambs and sills return boundary conditions to the 2-dimensional elements.

With the resulting information, a complete and detailed Bill of Materials can be produced see figure 39. Modifications of the 1- and 2-dimensional elements, such as the drilling of holes or the cutting of ends, as well as instructions for painting and assembly, are now also known. This enables the generation of a Bill of Work. Given the precise and detailed specifications, it is possible to use this information for online control of Computer

Numerically Controlled (CNC) machines and robots. This enables a high degree of production and assembly automation.

Item	#	length	width	colour
Panel_plywood	2	2600	1100	5
Panel_plywood	2	1900	1100	5
Panel_plywood	4	2600	1065	5
Panel_plywood	6	1900	1065	5
Panel_plywood	2	2600	1100	5
Panel_plywood	2	1900	1100	5
Panel_Glass	2	1900	350	
Panel_Glass	2	1900	315	
Panel_Glass	1	650	1100	
Panel_Glass	2	650	1065	
Jamb_P2	6	2700		
Jamb_P2	6	2600		
Sill_T5	1	4800		
Sill_T5	2	1150		
Sill_T5	2	2310		
Sill_T8	1	4800		
Sill_T8	1	1150		
Bolt_M8	12	30		
Bolt_M6	24	30		
Bolt_M6	25	25		
Parker_12	48	25		

FIG. 39.: This permits the automatic generation of a Bill of Materials for detailed costs analysis and production.

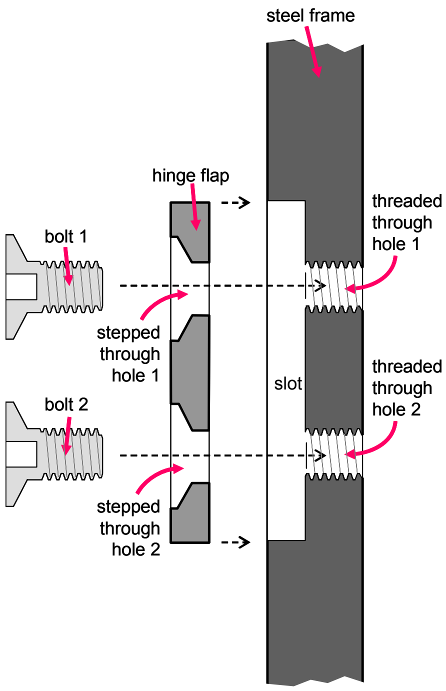


FIG. 40.: The door is connected with the door-frame using bolts. The connection is decomposed until the level of form features, which corresponds with elementary production processes. Bolts and steel frame are part of the wall frame, the hinge flap is part of the (moving part of the) door.

The lower part of figure 31 introduced the principle of network decomposition for a wall frame and a door. This is shown in more detail in figure 40. Figure 41 present a model of these connections in network view. In the decomposition structure, the bolts are considered to be part of the wall structure, while the hinge is considered to be part of the door. This is an arbitrary choice, but in product decomposition such choices have to be made frequently. Figure 40 shows that the detailed links between the bolt(s) and the holes in hinge flap and steel frame can be traced via nested ports, which are connected three levels higher in the hierarchy by a link defined by the Technical Solution for wall B.1. Through this principle, networks can be traced on different levels of detail.

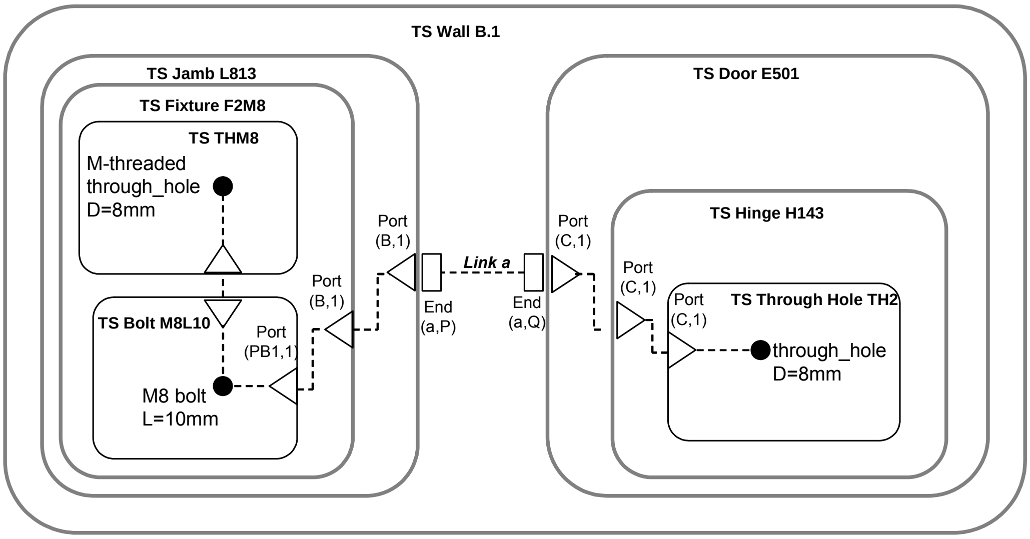


FIG. 41.: Decomposed Connectivity Network, showing the link between an M8 bolt and two through holes, one in a hinge and one in a doorpost, connecting on their turn a door with a jamb in wall B.1.

As the topology used for product definition is also seen as a kind of network – see section 5.4 - this means that also the topological relations between components and features can be decomposed and placed in a hierarchy. Given the fact that n-topology is independent of dimensional order, a feature such as a through hole can be represented as a vertex (point object), an edge (center-line) or a volume (the volume of material that has to be removed by drilling), see figure 42.

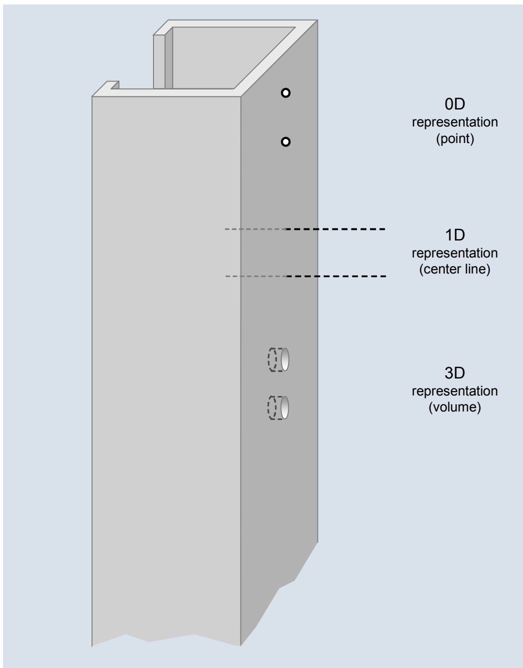


FIG. 42.: Three idealizations of a pair of through holes in a metal profile.

With the exception of the organisation-, staffing- and activity networks shown in figure 30, all principles described here were implemented and operationally used by the company. This implementation covered the specification hierarchy, a list - but not yet a hierarchy - of individuals, a library of parametric technical solutions of components and joints, and a 2-dimensional application of the topological principles. This application reduced the time for design and work preparation from 10 days to 0,5 day per project. It removed also the errors from the final specification. As a consequence, less material was needed at the construction site.

7.2 Roads and viaducts (1986-1993).

For the Dutch Directorate for Public Works and Water Management (Rijkswaterstaat), several applications were developed for the design and analysis of roads, viaducts and road signs. Emphasis was on parametric design and geometric modelling. For explanatory reasons, figure 43 shows a tiny part of a model of a viaduct. For each Functional Unit, such as the bridge pillar, the bridge deck, the road surface, the side walk and the guard rail, one or more Technical Solutions were available.

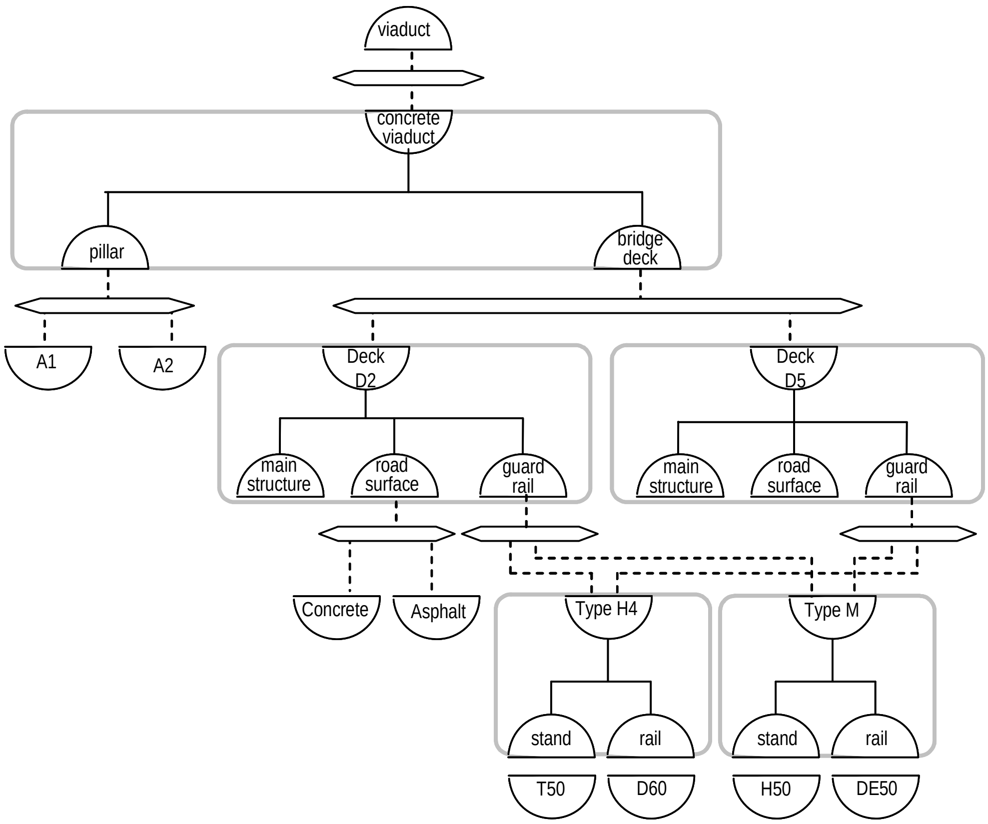


FIG. 43. Partial example of the Functional Unit - Technical Solution principle for the modelling of viaducts.

Each solution was described parametrically. These solutions were not restricted to discrete parts, but covered also connections and modifications of parts in order to be connected. For example, at the point where the bridge-deck of a viaduct rests on a pillar, the shape of the concrete deck and the positioning of reinforcement depend on the type of pillar on which it rests. This was solved by storing, for each possible combination of pillar and bridge-deck type, solutions for their connection.

How this principle works is depicted in figure 44. Figure 44a, on top, shows a simplified topological model of the viaduct. The vertices and edges refer to nodes in a network; each node corresponds with a Functional Unit of the viaduct. Edge E(1) represents the bridge-deck, and edge E(2) represents the pillar. The vertices in this model represent connections between bridge-deck and pillar (V(3)), between bridge-deck and ramps (V(1) and V(2)), and between pillar and ground (V(4)).

Please note that each edge is bound by two edges, which is an example of boundary-topology. However vertex V(3) is located <u>on</u> edge E(1) but does not bound E(1). This is an example of enclosure-topology (edge E(1) encloses vertex V(3)).

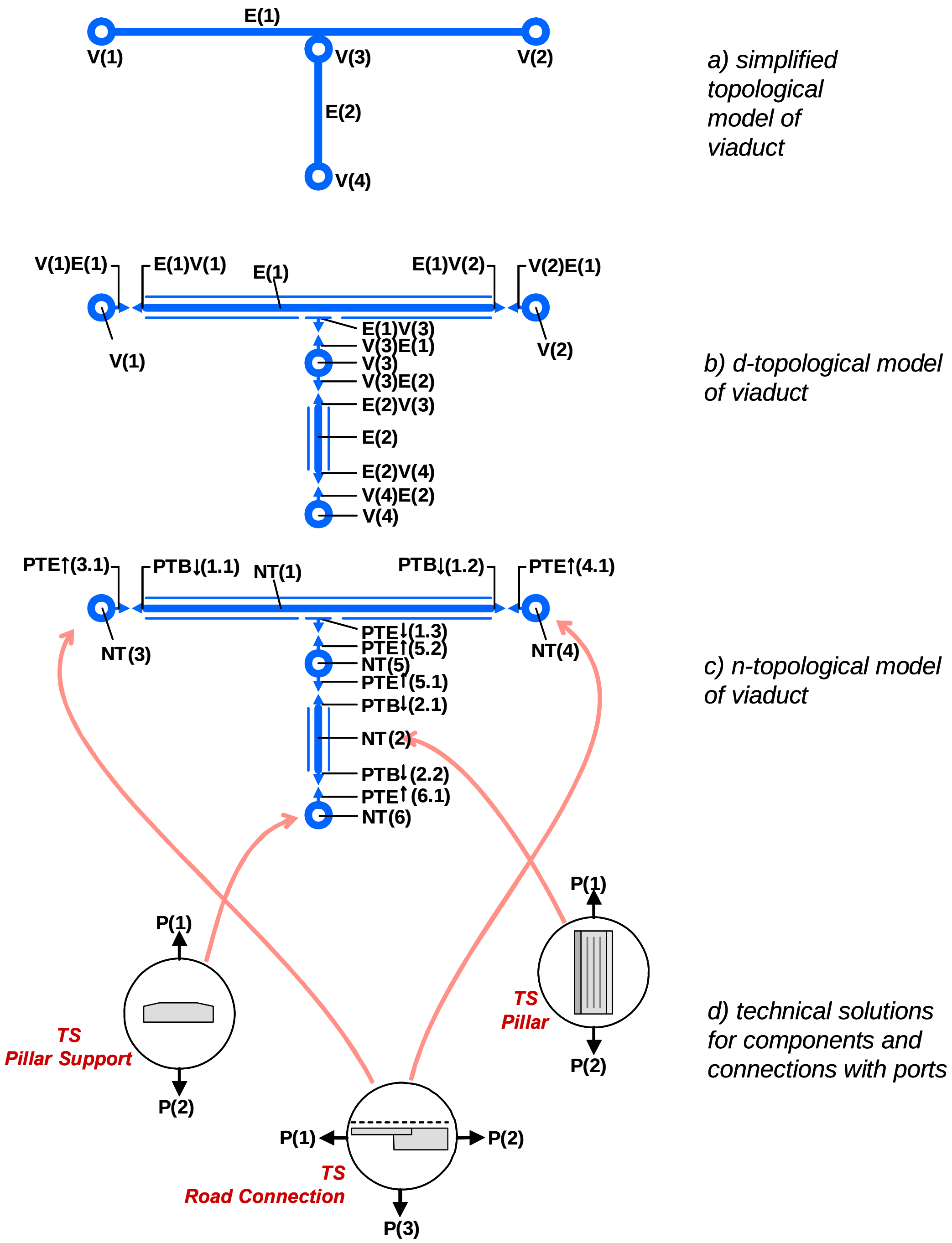


FIG. 44. High level topology of a viaduct, interpretation of d- topology as n-topology, and insertion of technical solutions for components and connections into the topological network.

In order to interpret this topological model as a network, two interpretation steps are needed. The first is the conversion of the non-manifold topology shown in figure 44a into d-topology. This is shown in figure 44b. The links between the edges and vertices are expanded here as edge-ends and vertex-ends. For example, edge E(1) has three ends: E(1)V(1), E(1)V(2) and E(1)V(3). Vertex V(3) has two ends: V(3)E(1) and V(3)E(2).

The second step is interpretation in terms of n-topology, which is independent of dimensional order. This is shown in figure 44c. The edges and vertices are interpreted as topological nodes (NT), and the connected vertex- and edge-ends as topological links.

Independent of the viaduct model, a library of Technical Solutions is maintained, which can be associated with Functional Units. Examples are parametric models of bridge-decks, pillars, and types of connections between them and the environment. Each Technical Solution has one or more Ports that enable it to be linked with other Technical Solution(s). Figure 44d shows only three Technical Solutions, one for the pillar, one for the pillar support, and one for connection of bridge-deck and ramp. The TS for the pillar has two ports, of which port P(2) is intended for linkage with the pillar support. The TS for the pillar support has also two ports, of which P(1) is intended for linkage with the pillar.

These ports can now be linked via the topological network in 44c. P(1) of the pillar support is associated with Port PTE↑(6.1), and P(2) of the pillar is associated with Port PTB↓ (2.2). Through this, the internal topological network of a Technical Solution can be connected with the internal topological network of another Technical Solution.

If the reference topology is changed in dimensional order, this will not affect the network, other than that it will be extended with more components or connections. For example, if the reference topology of the bridge is based on faces for bridge-deck and pillar, and edges for the connections, the n-topology shown in figure 44c will not change. It will however introduce new features. For example, if the bridge-deck is represented by a face, d-topology will introduce face-sides. The upper side of the face representing the bridge-deck may then be linked with a port of a Technical Solution representing the road surface.

The parameters of the bridge as a whole are transferred via the Functional Units to the Technical Solutions on a lower level. Thus, if high level parameters such as the span of the bridge or its width are changed, the dimensions of all other Technical Solutions on the lower level are automatically changed too. This transfer is based on algorithms. For example, the parametric model of bridge deck D2 incorporates the rule that its width is 1.2 times the width of the road, while the model of pillar B incorporates the rule that its width is 0.9 times the width of the bridge-deck.

For each Technical Solution, boundary conditions and rules for their application are given. Pillar A1, for example, carries weight up to 240 tons and has a maximum length of 7meter. If a selector combines this Technical Solution with a Functional Unit of which the requirements exceed these limits, the Technical Solution returns information to the selector that this FU/TS combination is invalid.

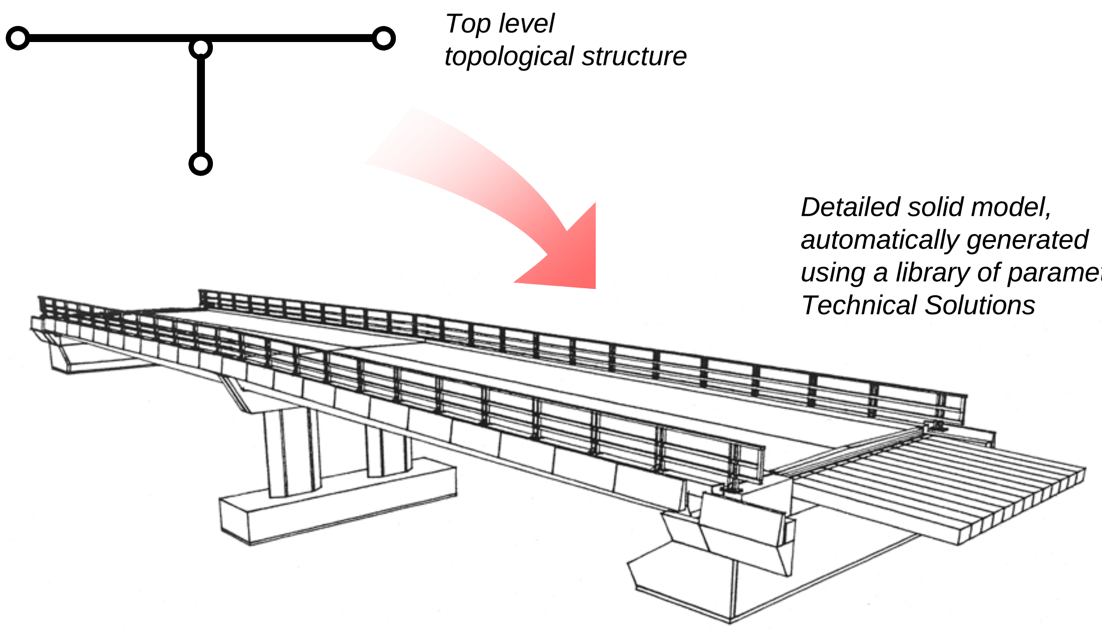


FIG. 45. Solid model of a viaduct; automatically generated from its parametric model.

If no Technical Solutions are found that meet the requirements of certain Functional Units, the designers can modify the Solutions or add new ones to the library. The method enables the modelling of a broad family of concrete viaducts with different configurations.

Figure 45 shows a picture of a detailed solid model that was automatically generated from the parametric model, using the described technique.

7.3 Design, analysis and computer controlled production of ship propellers and sheet metal parts for aircrafts (1988-1992).

The European research project IMPPACT (Integrated Modelling of Products and Processes using advanced Computer Technologies) aimed at the integration of design, analysis and CNC production automation applications of complex shaped parts and sheet metal parts. It resulted in the most complete implementation of the theory so far.

In this project about a dozen different computer applications were integrated via a shared database. This integration was based on the use of parametric features, according to the principles outlined in this article. Only for a few applications it was necessary to exchange explicit geometric models.

One of the demonstration lines that will be presented here is that of ship propellers. The company LIPS (nowadays part of the Wärtsilä group) was - and still is - a leading manufacturer of large ship propellers. Propellers are usually unique, one-of-a-kind products, specifically made for each individual ship. Their shape depends on the flow of water around the ship hull, the intended cruise speed, the power of the engine, requirements for noise or vibrations, and many other factors. Hence, the shape follows logically from functional parameters. Design does not start with geometric modelling. In a first stage the ideal shape of the centre plane of each blade is calculated, then, after calculation of required strength with a Finite Element application, thickness is added, resulting in a volumetric model. This model is once again analysed and corrected for strength and cavitations (vaporized water bubbles, caused by the low pressure on the suction side).

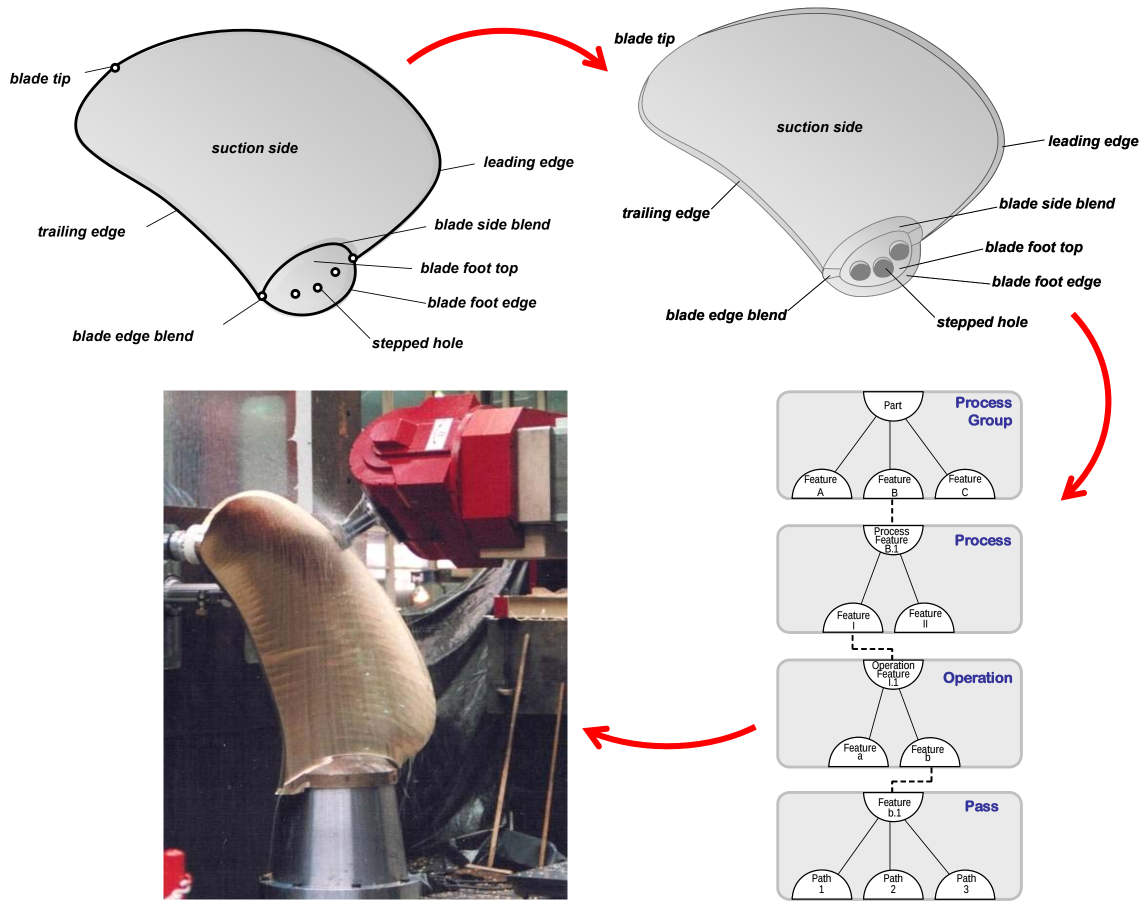


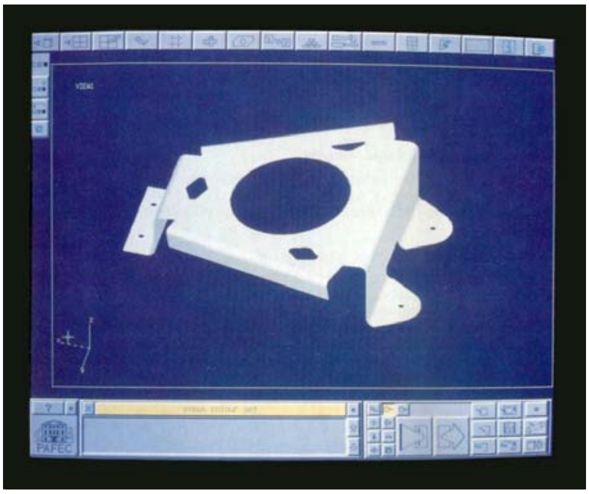
FIG. 46. Application of the theory for the design and computer controlled production of propeller blades.

In the IMPPACT project, the entire shape of the blades was defined with parametric features, such as for the pressure and suction side, the blade edges, the blade foot, and so on. Although a distinction was made between design and production features, reflecting different views on the propeller blade, it appeared that these views could be combined in a single specification hierarchy. Four decomposition levels were identified, matching the organization of production activities as process groups, processes, operations and passes. A pass was a single movement of a tool.

The upper left picture of figure 46 shows the topological arrangement of skin features. In the upper right figure the blade geometry is made explicit by transforming the 0D and 1D topological nodes (i.e. vertices and edges) into 2D topological nodes (i.e. faces). This transformation was supported by the application of n-topology, such as described in section 5.4.

The lower right diagram shows a portion of the specification hierarchy of production features that correspond with material removal processes. An operation is defined here as a sequence of passes with a single tool, and a pass is defined as single movement of the tool. The photo on the lower left shows the CNC controlled milling of a propeller blade.

The same principles were used for the design, work preparation, production planning and production of sheet metal parts. Parts are cut from a metal plate with a router or a (laser) cutter, holes are produced by punching or nibbling, the flat part is folded into its final shape through bending, then the edges of the part are deburred, the surface is finished or polished, and finally the part may be coated. Production planning and -optimization happens in reverse sequence of the actual production process: its starts with the designed geometry of the finished part, then the part is unfolded, features such as holes and slots are removed, a circumference of the final geometric contour is calculated for the routing path, and finally multiple parts are arranged such that they can be produced from one large piece of sheet metal with minimal loss of material. The latter process is called nesting. All these intermediate stages of production, including process related geometry, can be modelled with the principles in the present theory.



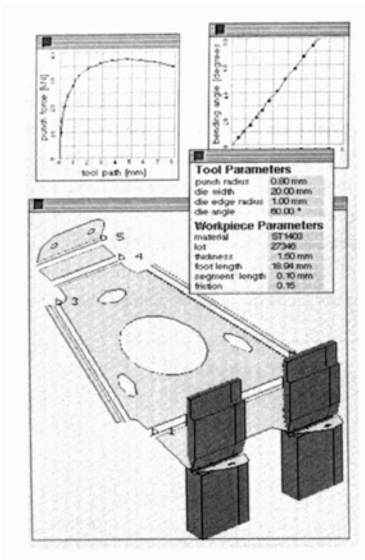


FIG. 47. Screenshot of a designed sheet metal part (left) and work preparation for bending (right)

Of particular importance is that, during production, parts must have additional features to support handling in the semi-automated production process: each sheet metal part has at least two rivet-ears so that it can be fixed during bending, surface finishing and/or coating. These rivet-ears, which are not part of the initial design, are removed in the last stage of production. It may also be that parts are grouped and connected so that they can be processed jointly. This object is then split into parts after processing. The requirements for temporary features and the splitting and combination of objects are met by the process modelling principles such as described in section 4.3.

More details about this application can be found in (Gielingh and Suhm 1993).

7.4 Design, construction, maintenance and operation of 29 Process plants (1996-…).

The next case that will be described concerns the design, construction, maintenance and operation of 29 process plants in the oil and gas sector.

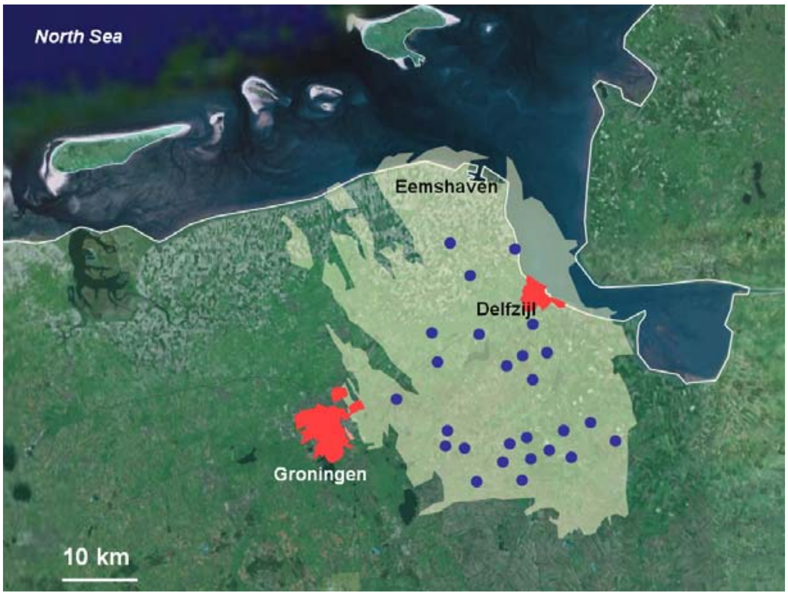


FIG. 48. *Northern part of The Netherlands with the towns Groningen and Delfzijl. The light green area is the Groningen gas-field. Locations of gas production units (so called clusters) are marked with blue dots.*

Below the surface of Groningen, a province in the northern part of the Netherlands lies one of the largest reservoirs of natural gas in the world. This reservoir is exploited since 1958 and is now more than half empty. As the natural gas-pressure has dropped there was recently a need to install compression units. Also, as the installations were nearing the end of their lifetime and had high operational costs, there was a need to renovate them. The company that exploits this gas-reservoir - NAM, a joint venture of Shell and Exxon - decided to contract this huge effort as an integrated Design-Build-Maintain project. The project started in 1996 and has a duration of at least 25 years.

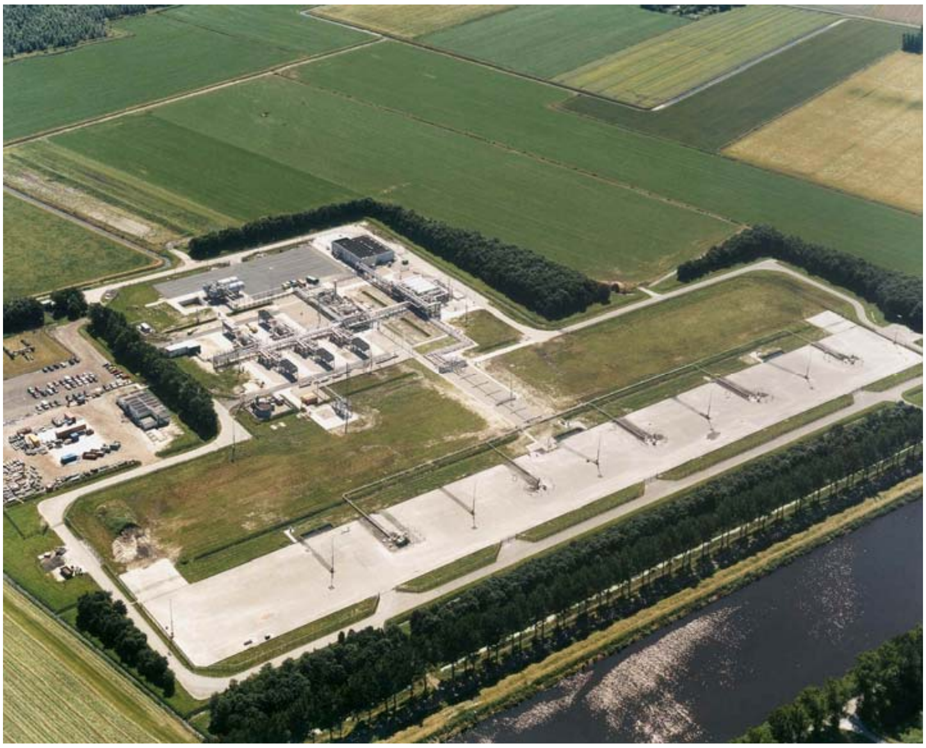


FIG. 49. *One of the 29 gas production clusters. The wells surface at the light-grey rectangular area. The gas is then treated in a plant (centre) before it is supplied to the gas distribution network.*

The natural gas is exploited via hundreds of pipelines that reach the surface of the Earth on 29 locations, called clusters, distributed over a large area of land; see figure 48. Each cluster is equipped with a small plant for the drying and cleaning of the gas, and for the separation and processing of pollutants. An aerial photo of one of these clusters is shown in figure 49.

The 29 plants could not be constructed all at once. In the most favourable scheme between 2 and 3 plants per year could be constructed. Consequently, construction of the last plant would start 12 to 15 years after the first one. In these years, construction-, operation- and maintenance personnel could gain a lot of experience that could be used to improve the quality of the design, the quality of processes, and the reduction of overall lifecycle costs.

An important tool for continuous improvement was a knowledge feedback system, see figure 50. Knowledge created in each process would be used by that process for improvement. But this knowledge was also made available to the design and planning disciplines, so that design and planning could be further optimized. The latter is also called *front loading*.

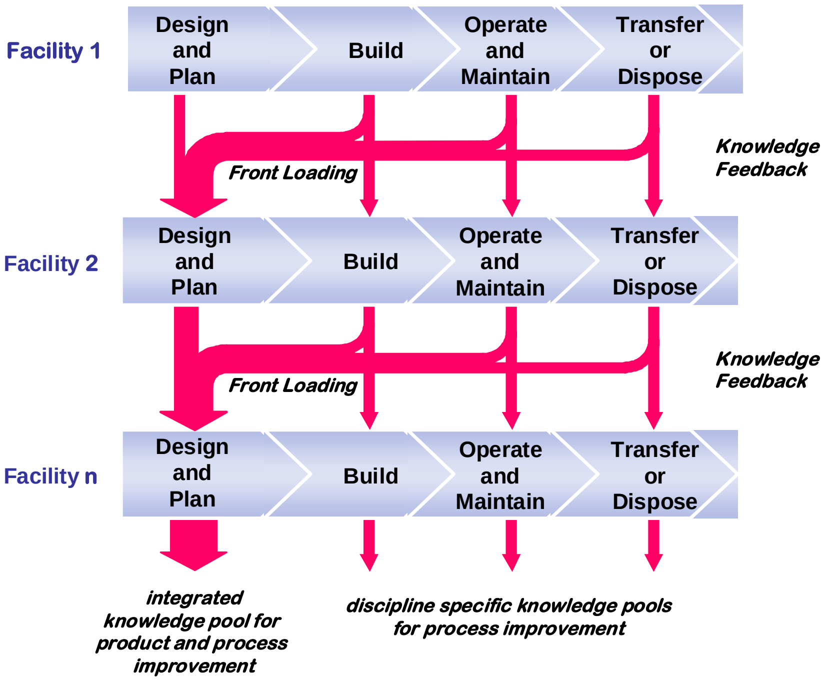


FIG. 50. *Principle of the knowledge feedback system. Lifecycle knowledge acquired during the construction, maintenance and operation of each plant becomes available to each discipline and is collected in a central data warehouse for use in future design projects .*

New knowledge could lead to design and planning changes. In order to avoid that a simple design change would propagate too far in other places of a design, modularization principles were applied: this meant that older Technical Solutions could be replaced by new ones.

Design changes could however have disadvantages for operation and maintenance, in particular if all plants become different. Hence, it was decided to strive for an optimum between functional uniformity (i.e. standardised Functional Units), and technical and technological differentiation (i.e. to allow different Technical Solutions for each Functional Unit).

Lifecycle experiences were collected per Functional Individual and Technical Individual, to support improvement of building and maintenance processes.

Unlike the applications mentioned before, the principles were not implemented here in the form of a parametric product model, but as a method for the structuring of plant information. This method was supported by a Product Data Management system. The resulting information structure was essentially the same as sketched in figure 20, with the only difference that the Generic level (i.e. the parametric product description) was missing. This was because most CA-applications were not parametric. However, the knowledge feedback loop from Actual to Imaginary was implemented and 'short-circuited' via the Specific Level; see figure 51.

Most information was retained in the Specification hierarchy. For example, if clusters 1-6 had the same gas drying units, they shared one specification. If clusters 7 - 18 were equipped with a slightly different one (version 2) and clusters 19-29 again a different one (version 3), then there were in total only three different 'Technical Solutions' for this Functional Unit, one for each version. If all 29 gas drying units have the same support structure, independent of the version of the unit of which they are part, there was again only one Technical Solution that described it. This independence between wholes and parts was made possible by the Hamburger concept.

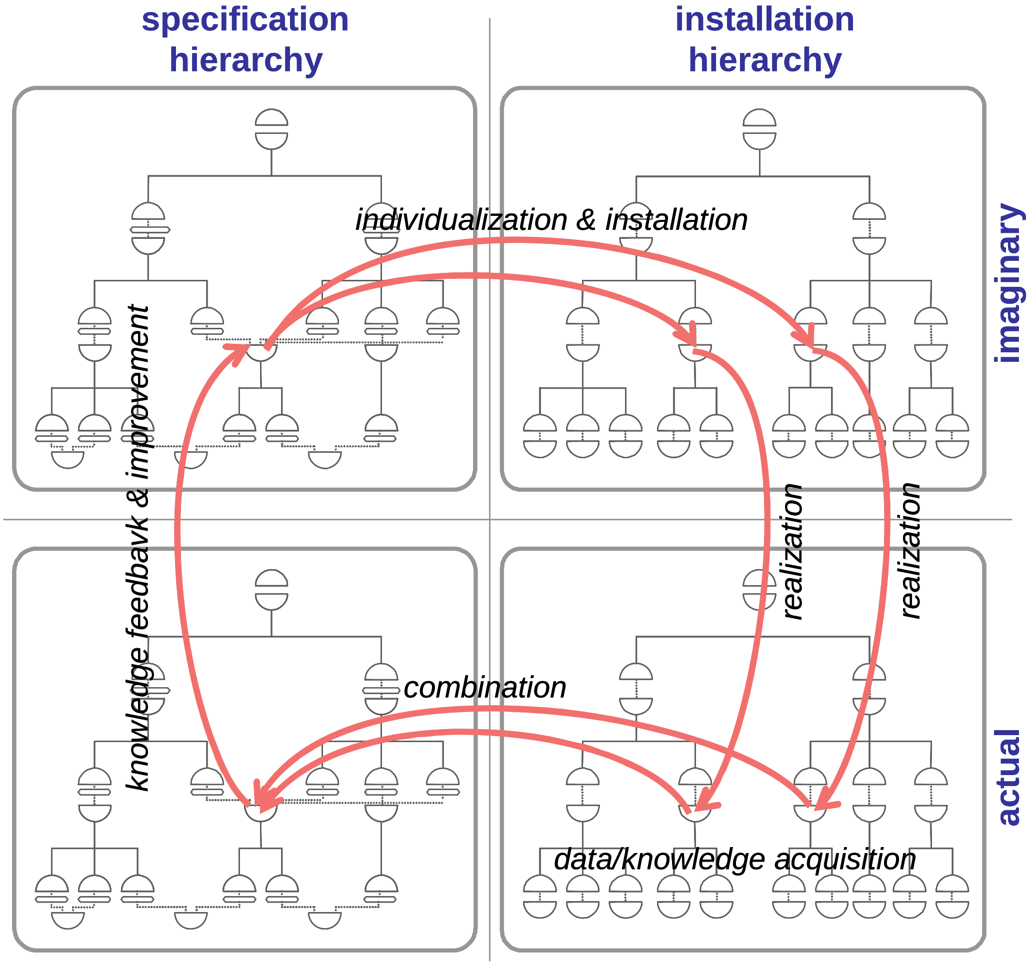


FIG. 51. *For practical reasons, the knowledge feedback system for the Groningen Long Term project was implemented only at the Specific and Individual/Occurrence level The ideal situation is sketched in figure 20.*

For the collection of data input in the actual plants (the right-lower corner of figure 51), three types of data and knowledge sources were identified; see also figure 52. The first is automated data collection from sensors and other equipment in the plant. The second is non-automated data collection such as from inspection reports. Inspection reports are directly entered as data in a computer, such as lap-top or a hand-held device. These two sources of data were still raw and needed to be processed before they became useful.

Figure 52 shows that this was a two-stage process. Raw data was analyzed and diagnosed for operational usage. Not all of that data was useful for other purposes. The filtered data was stored for long term data analysis, such as for tactical purposes (maintenance planning and scheduling) and strategic purposes (continuous improvement).

Tactical analysis was supported by a knowledge system, using rule based inference, while tactical analysis was supported by data mining technology.

The third source, which required less processing, was explicit knowledge recording, such as in the form of idea's and suggestions for improvement.

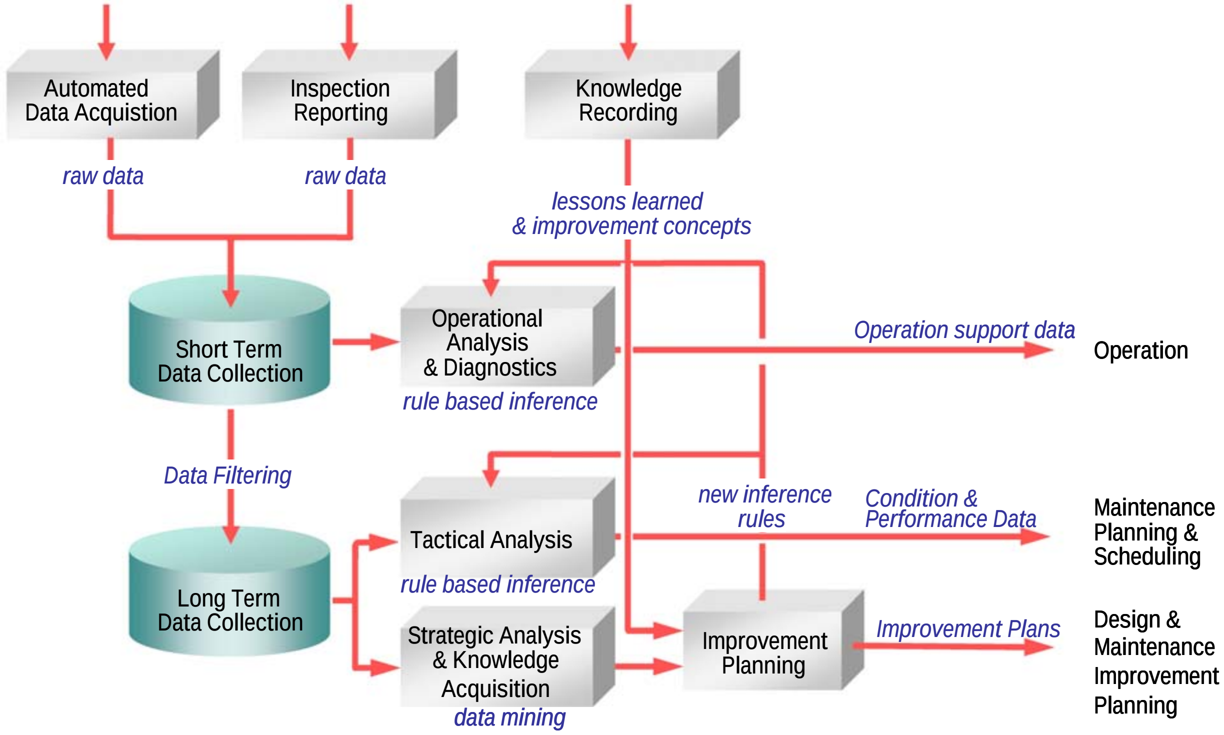


FIG. 52. *The GLT project used three principal sources of data/knowledge acquisition (top) that were analyzed and processed for operational, tactical and strategic usage.*

The various kinds of data and knowledge associated with design modules were stored in a common data warehouse and managed with support of a PDM system.

Apart from the PDM system, a large number of other computer applications were used in this project, ranging from a variety of CA-applications, ERP, maintenance management and operation management systems. The logical knowledge structure as described in this paper affects most – if not all – applications in use.

Practical limitations made it however unviable to change all applications according to the principles outlined in this paper. The reason was that software changes had to be done in a fully operational environment, which would disrupt the ongoing work too much. Therefore the principles were mainly applied as a working practice within the organization, supported by the PDM/WFM software.

Despite this restriction in software support, the principles appeared to be highly beneficial for the structuring and organization of knowledge and supported continuous improvement of the design, construction and servicing processes. Benefits result from cost reductions and higher end-user value. Lifecycle costs were estimated to be reduced between 25 and 30%, while the system also contributed to other performance factors such as higher availability, better reliability and increased safety (Gielingh 2005).

6.5 Related applications and research

There have been many other applications, including ones in which the author was not personally involved. Of these, a few will be mentioned here.

In a study for the Netherlands authority for government buildings, A.de Scheemaker (1991) showed that various principles could be used for the modelling of organisations. She developed a model of building performance with respect to the housing of government organisations that fits seamlessly with GARM based building models.

P.Willems (1998) showed that many of the principles can be used for the modelling of roads, road artefacts (such as viaducts) and road furnishing. He was the original inventor of the topological principles mentioned in the present paper, and applied them to road modelling.

S. van Nederveen (2000) applied the principle of Object-trees in a large construction project for a high speed railway in the Netherlands. Such a large project is divided into many small 'sub-projects' or 'sub-problems' such as the interfacing with existing infrastructure. These interfaces require the construction of bridges, tunnels or noise barriers. The method is of particular importance for the interaction between disciplines and actors, so that the number of interdependencies, requiring time for their resolution, can be minimized.

S. Özsariyildiz (2006) developed a method for Inception support of large scale construction projects, including a case study for power plants. It contains a method for optimizing the search of Technical Solutions for a large set of requirements on several levels of problem decomposition.

A closely related study was done by Jain and Augenbroe (2003) in which the authors describe a method for the automated search for solutions in electronic catalogues using multiple performance indicators.

STABU Lexicon (Woestenenk 2004) is an electronic version of the STABU standard reference specification, of which the latter is widely used in the Netherlands. The Lexicon, which is an ontology of building elements, components and processes, makes use of the Functional Unit / Technical solution principle.

R. van Rees (2007) developed a web-based solution for the transaction of knowledge between construction companies. It uses a building-construction ontology, bcoWeb, which is also based on the Functional Unit / Technical Solution principle, and tools that support the communication between 'smart' computer applications.

7. ACKNOWLEDGEMENT AND CONCLUSIONS

7.1 Acknowledgement

Of the many people who made major contributions to this theory I would like to thank in particular Peter Willems, Peter Kuiper, Sander van Nederveen, Michel Böhms and Frits Tolman (former colleagues at TNO), Jeff Wix, Barbara Warthen and Bill Danner (STEP), Alexander Suhm (co-editor of the IMPPACT reference model).

7.2 Conclusions

This paper describes a theory for the modelling of complex systems, such as buildings, civil works and plants. It addresses not only static but also dynamic aspects, and is capable to support a full system lifecycle. The theory is not only applicable to large systems but also to parts and features of parts.

Unlike the original publications, the theory is not presented here as an information model, conceptual schema or ontology. The meta-models of widely used modelling languages, such as Express, UML and OWL make assumptions that are sometimes in conflict with the presented theory. The theory may actually be considered as an alternative 'meta-model' but is, do its complexity, less suited for direct use as a modelling language. Despite this, all elements of the presented theory have been implemented and applied at least once in projects of different kind. Some of its applications were used in industrial practice; other applications were of academic nature. These applications and implementations gave valuable feedback and led to improvements of the theory. This article presents an overview of its current state.

A proposed new Theory of Notions (Gielingh 2008a) should remove the abovementioned conflicts between the 'extended GARM' and modelling languages. As this new theory is not yet an applicable method, it is not used in the present paper. This will be subject of future work.

8. REFERENCES

Barnes, J. (1984) *The Complete Works of Aristotle*. Princeton University Press, Princeton NJ, ISBN 0-691-01651-8

Bateman, J.A., Henschel, R., Rinaldo, F. (1995) *The Generalized Upper Model version 2.0*; www.fb10.uni-bremen.de/anglistik/langpro/webspace/jb/gum

Björk, B.C. (1994) *RATAS Project—Developing an Infrastructure for Computer-Integrated Construction; J. Computation in Civil Engineering*., Volume 8, Issue 4, pp. 401-419.

Björk, B.C. (1995) *Requirements and information structures for building product data models*; VTT Finland, ISBN 951-38-6030-2.

Bjørke, O., Myklebust, O. (1992) IMPPACT, *Integrated Modelling of Products and Processes*. Trondheim, ISBN 82-519-0973-2.

Foliente, G. (2005) *Performance Based Building R&D Roadmap*; International Council for Research and Innovation in Building and Construction Development (CIB), Rotterdam, ISBN 90-6363-048-4. Also available at www.pebbu.nl.

Gielingh W. (1986). An AEC reference model based on a relational reference representation; ISO TC184/SC4/WG1 Document N77.

Gielingh W. (1988a) *General AEC Reference Model (version 4)*; Document N329 of ISO TC184/SC4/WG1, Oct. 1988. Published as TNO Report BI-88-150.

Gielingh W (1988b). *General AEC reference model (GARM) an aid for the integration of application specific product definition models*, in: Christiansson P, Karlsson H (ed.); Conceptual modelling of buildings. CIB W74+W78 seminar, October 1988, Lund. Also available at http://itc.scix.net/cgi-bin/works/Show?w78-1988-165.

Gielingh W. and Suhm A. (1993). *IMPPACT Reference Model for Integrated Product and Process Modelling*; Springer-Verlag, ISBN 3-540-56150-1 / 0-387-56150-1

Gielingh W., Los R., Luijten B, Putten J.van, Velten V. (1996*) The PISA project, a survey on STEP*. Shaker Verlag Aachen. ISBN 3-8265-1118-2.

Gielingh W., (2005). *Improving the performance of construction by the acquisition, organization and use of knowledge*. Delft. ISBN 90-810001-1-X.

Gielingh W., (2008a). *Theory of Notions and Knowledge DNA: a new Paradigm for Semantic Interoperability of CA-Applications.* Journal of Computer Aided Design (submitted August 2006, to be published in 2008).

Gielingh, W. (2008b). *Cognitive Product Development: A Method for Continuous Improvement of Products and Processes*; Journal of Mechanical Engineering 54(2008)6 page 385-397, ISSN 0039-2480 (also available at http://en.sv-jme.eu/current-volume/jme-54-2008-6/).

IAI/IFC (2007). www.iai-international.org

Jain, S., Augenbroe, G. (2003) *A Methodology for Supporting Product Selection from E-Catalogues*; ITcon Vol. 8 (2003), pg. 384.

Kemmerer S. (1999). *STEP, The Grand Experience*. NIST SP 939.

Kim J., Pratt M., Yver R., Sriram D. (2008*). Standardized Data Exchange of CAD Models with Design Intent*; Journal of Computer Aided Design, Volume 40, Issue 7, Pages 760-777.

Luiten G., Froese T., Björk B.-C., Cooper G., Junge R., Karstila K. and Oxman R. (1993). *An information reference model for architecture, engineering, and construction*. In: Management of information technology for construction (eds. Betts M. and Tham K.). Singapore: World Scientific & Global Publication Services

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A. (2003) *Wonderweb Deliverable D18*, IST Project 2001-33052.

Nederveen, S. van (2000) *Object Trees - Improving Electronic Commuciation between Participants of Different Disciplines in Large-Scale Construction Projects*; Delft, ISBN 9013619-3.

Neisser U. (1976). *Cognition and Reality - Principles and Implications of Cognitive Psychology*. New York. ISBN 0-7167-0477-3.

Özsariyildiz, S. (2006); *Inception support for Large Scale Construction Projects*; Delft, ISBN 90-808007-2-4.

Pattee H. (1973). *Hierarchy theory - the challenge of complex systems*. Braziller, New York. ISBN 0-80760674-X.

Pratt, M. (2007). www.mel.nist.gov/pptb/features_bib/ features_bib.html

Rees, R. van (2007). *New instruments for dynamic Building-Construction: computer as partner in construction* (thesis), ISBN: 978-90-808007-3-1 (also available at http://vanrees.org/proefschrift.pdf).

Scheemaker, A. de (1991). *Prestatiebeheersing in het Rijkshuisvestingsproces*. Delft, ISBN 90-6275-692-1.

Smith E. (1988). *Concepts and thought*, in: The psychology of human thought. Cambridge University Press, Cambridge. ISBN 0-521-32229-4.

Smith, B., Grenon, P. (2006) *The Basic Formal Ontology (BFO)*; www.ifomis.org/bfo/1.1#

Szigeti, F., Davis, G. (2005). *Performance Based Building: Conceptual Framework*. International Council for Research and Innovation in Building and Construction - Development Foundation. ISBN 90-6363-051-4.

SUMO (2007) *Suggested Upper Merged Ontology*, at: www.ontologyportal.org.

Turner, J. (1988) *AEC Building Systems Model*, ISO TC 1184/SC4/WG1, Document 3.2.2.4.

Willems, P. (1988) *A Meta-Topology for Product Modelling*, CIB W78 conference, Lund, October 1988. http://itc.scix.net/cgi-bin/works/Show?w78-1988-213

Willems P. (1998). *Conceptual Modelling of Structure and Shape of Complex Civil Engineering Projects*. Leidschendam; ISBN 90-9011888-8.

Wix, J., Tolman, F., Poyet, P., and Björk, B.-C. (1994) *Building Construction Core Model*, Document N341, ISO TC184/SC4/WG3.

Woestenenk, K. (2004) *STABU Lexicon*. (http://www.stabu-lexicon.nl/)