Theme: Closing speech

Title:

Model-based Architecture - an Architects View

Author(s): Assoc. Prof. Kristian Agger
Institution(s): Aarhus School of Architecture -AAA, Centre for Integrated Design -CIB
E-mail(s): kristian.agger@a-aarhus.dk

Keywords: *Dynamic model. Detailed design. Integrated design-production. Implementation in Architecture*

Introduction

Model-based Architecture is still far from being a concept widely adopted in the building industry according to a non-published working paper [23]:

> The industry as a whole is now on the **threshold of adapting** the actual concept of interoperability and slowly putting product models into use. The **users now can be described as early adopters** and **the majority of the industry (estimated at over 90%) has not reacted yet**. The potential in terms of concrete cost savings is visible, but the actual impact on the processes and the businesses is not known yet.

Why is it so?

The **architects are first in the chain** as they are creating the initial model in the beginning of the design process. That is why it is important that architects feel that Model-based Architecture is attractive.

But are the architects too conservative?

> You can hear architects with only little knowledge of modelling saying, “We don’t need models, we have the model in our head”.
>
> On the other hand the architect who has an insight in modelling would say, “We will not use models before we can model efficiently in full detail”.

Or is it the quality, the capacity and performance of the models used, and thus the modelling tools offered to the architects that is the problem?

Recent organisational research [19] point out that insecurity especially in building is an important factor, and that only spontaneous, non-linear and dynamic design <u>and</u> implementation leads to success.

Following this we could ask the question: “Is the product models offered to Architecture supporting spontaneous, non-linear, dynamic and design and implementation?”

		Design	
		Planned Linear Static	Spontaneous Non linear Dynamic
Implementation	Planned Linear Static	Fiasco	Fiasco
	Spontaneous Non linear Dynamic	Fiasco	Success

The definition of a model and of interoperability in [23] goes like this:

> With “building product model”, one normally refers to the data used in a construction project or a particular type of product model, a computer interpretable description of a building, structured according to some building product data model
>
> Interoperability means that **a building product model data can be transferred to or shared with** all of the relevant applications, throughout the life cycle of a facility or a building, **by using a common logical structure**.

This means, that a model is structured data, that in order to support interoperability has to use a common structure. This model definition defines a pure data model, and the dynamic aspect is not very present.

Nearly all models found in applications and for transfer is based on the fundamental idea that: **A building is an assembly of building elements**. Could this be the reason why we have not been able to develop models and modeling tools that have been accepted in architecture the same way that drawing tools have?

Is it impossible to make an element-based model that is detailed and dynamic?

An alternative fundamental idea to decomposing a building for modeling is: **A building is a collection of interrelated spatial elements that contains constructions or functions**. This space based structure could lead to modeling tools that support architecture better.

Author background

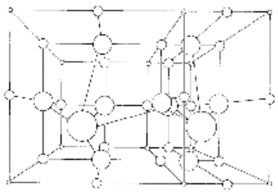
My interest in IT started back in 1969 as a student of architecture at the Aarhus School of Architecture, where I attended my first programming class at Aarhus University. The programming language was ALGOL a forerunner for Pascal.

Hemispherical fisheye perspective projection.

The fist project we made was a computer program for fisheye perspectives. The perspectives as well as fisheye slides were projected to a vertical hemispherical -2 meter radius- screen, and thus formed a kind of early “cave” experiment.

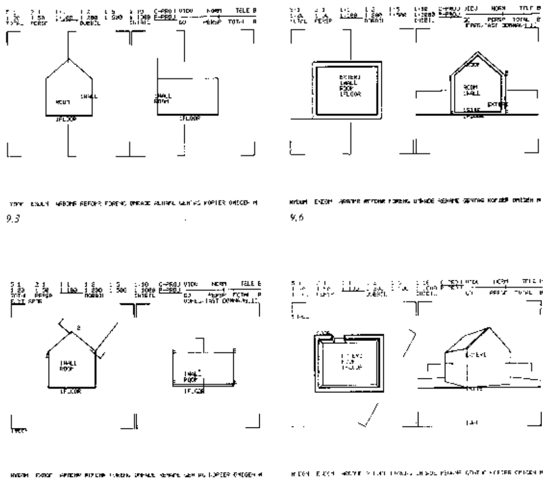
Following 1½ semester of studying computer science at Aarhus University I made my graduation project at the School of Architecture.

Datastructure for the function and geometrical description of buildings.



Data structure for the function and geometry of buildings

This project started as a graduating project in 1972-73 and continued as a post-graduate study in 1973-76 [2]. In the project a modelling system was implemented on a mainframe using Pascal and Tektronix screen. The system data structure was based on spatial domains containing constructions or rooms. The domains were geometrically defined by surfaces common to and thus relating the neighboring domains. The system had an interface with two orthogonal projection plans. An algorithms for cutting and hidden line was implemented. Unfortunately the system did not survive the death of the mainframe computer.



Photogrammatry

Three versions of a system for analytic photogrammetry, measuring and plotting, were developed in the periods 1977-84 and 1994-95. The last version was based on the database system FileMaker and the CAD system MiniCAD.

Architectural design with computers. Gable and Scribe

In mid 80es a group of Danish architects travelled around in the UK to visit GDL, Acropolis, Gable, CADdraw, Rucaps and Scribe, all long forgotten. GDL and Acropolis was tried out in Danish architectural offices. And a year later Gable was tried out at an architectural office in Aarhus. Experiments from the pilot project was reported by Aarhus School of Architecture [4], with the conclusion that Gable had the potential as a modelling tool as well as a drafting tool.

At the same time a strategy combining Scribe as a sketching tool with AutoCAD as a tool for detailed design was introduced in education and in architectural practice. But in the following years the practice gave up modelling, and AutoCAD gained ground as a drafting tool.

Facilities Management. EjdInFo

From 1991-95 a prototype system for facilities management called EjdInFo was developed and implemented for the Ministry of Housing. The system had a graphical interface to the objects and this was reflected in the structuring of the FM objects.

ArchiCAD

During the 90es Zoom and later Form-Z became the basis for modelling in education. But in the mid-90es special building modelling classes based on ArchiCAD started.

And ArchiCAD did not only serve as a building modeling prototype system in teaching, but also became an important source of practical modeling inspiration in the following research.

Geno project

In the Geno project "Geno-Object-Classes in Construction IT" [14] a new geometrical structure for the building product model was invented –see the below heading "Partecture, designed for dynamics and details".

This was based on the data structure developed in the early "Datastructure for the function and geometrical description of buildings" project [2] –see above, and was an extension of this to allow effective detail modeling.

In the Geno project a strategy for modelling tools called Modellors was introduced. The Modellor concept and along with this the Geno concept have been extended and refined during the last three years in the Modellor project. In the last year we have supplemented the research with an IFC Pilot Project, a practical model transfer via IFC format [24]

Modellor project

The outset for the Geno and the Modellor project is a deep concern for the situation that modeling is not adapted in architecture.

The following is a report on ideas for changing this. It is not a clear academic report, as I have not been trained as an academic in my education.

So the method used in this work is more the method of an architect.

Spontaneous, non-linear, dynamic. With the drawback and advantage this have.

IT development in the building sector

The building industry, at least in Denmark, is subject to both organizational and structural changes, to become as productive as other similar industries. IT in construction will develop to support this but will also develop in its own right as IT hardware and software techniques are improved. The development will follow the overlapping phases:

On a short view ***Traditional design*** is developing cooperation via internet portals, where all participants have equal access to traditional project documentation.

On a medium view ***Model-based design*** is about to take over in design practice. Modelling is based on parametric components but uses traditional project documentation. This is possible today, but architects are conservative and unwilling to skip orthogonal projection and floor plan as the main medium for design documentation.

On a long view ***Model-based construction*** will be implemented in the whole sector and integrate design and construction through a standardized/comprehensive model - the Partecture. Modelling will be based on intelligent modelling-tools, - Modellors - specialized to a building-part or – component system and representing it in the model in a "smart" dynamic way. Evaluation, a holistic view of a certain aspect of the building, is performed with software modules - Evalors - with built in engineering etc. knowledge.

In this development the change from traditional documentation to model-oriented methods is a very deep cultural change, especially for the architects, and the implementation in the whole industry cannot take place as an evolution. So to gain the full potential benefit of IT in the building-industry takes a very intensive educational effort, but also an improvement of the IT tools supporting "Model-based Building".

The following describes ideas – inventions - for the next generation of building IT, an idea-development that is based on the supposition that IT can support and make possible some of the organizational changes that will improve the quality and productivity in the sector, that is:

***Holistic management* -** enables one or a few architects to monitor and manage the whole design- and building-process.

Late (parallel) ***decision making* -** makes it possible to make decisions when the best basis exists, and only when it is absolutely necessary in a linear process. It also makes more parallel decision-making possible.

Integration of design and production: enables cooperation with joint responsibility via the commonly shared data-model. This leads to the following superior demands for the next generation of IT-building systems:

Minimal data-redundancy: data principally exist only once, even when it comes to integrating detailed design in the model.

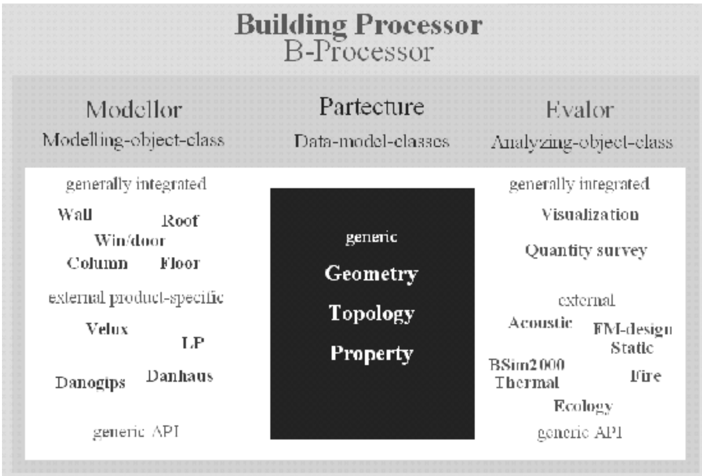
Maximal data-integration: relations between data are expressed and constantly updated

Maximal data-dynamic: data are updated automatically if demanded by changes in related data.

Minimal information-flow: Handbook- and Product-information is incorporated in the system of the evaluation and modelling tools.

IT-system for Building

The IT system for Building consists of – and integrates – three different parts: the data-model, a suite of tools that create and maintain the data-model, and a group of subsystems that evaluates aspects of the model.



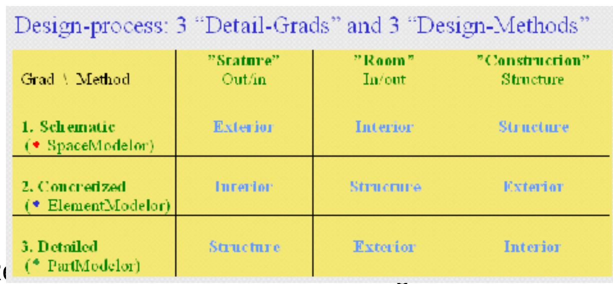
As a concept the IT building system is not well defined – and does not even have a generally accepted term associated, so in the following the term ***Building-Processor*** or short ***B-Processor*** will be used.
The data-model of the B-Processor has been called many different things – product-model, virtual model, digital model etc.- therefore the term ***Partecture***, a combination of the Danish/Italian term “partitur” (UK musical score) and architecture will be used in the following. The Partecture is the generic structure of data-object classes necessary in the heart of the B-processor to allow for data-exchange and multi system plug-in development. The IAI/IFC is so far the only development of a proposal for a common Partecture in the industry.

A modelling tool in the B-Processor will in the following be termed a ***Modellor***. Each Modellor is specialised to model a specific part of the building: a functional space, a building part, a building component/meta-component or even as a plug-is developed by the building component producer, a specific building product. The Modellor represents the model-part in the Partecture and has a built-in intelligence so that it adjusts dynamically to changes in related Modellors in the Partecture.

The third part of the B-Processor is subsystems that perform a holistic evaluation of an aspect of the performance of the building. This part will be termed an ***Evalor***. The Evalor extracts data for the Partecture and produces a picture, a price, the energy-costs, etc. It is beyond the scope of this project to deal with specific Evalors, but the development of ideas for the Partecture takes the Evalor concept into consideration.

Design Process and Method

Very technically speaking designing can be regarded as the process of dividing the total project space into spaces for constructions and spaces for functions, and to define what the contents of each cubic millimetre of these spaces are: solid, fluid or gaseous. But the solutions to that are numerous, so there must


Design-process: 3 “Detail-Grads” and 3 “Design-Methods”

Grad \ Method	"Srature" Out/in	"Room" In/out	"Construction" Structure
1. Schematic (* SpaceModelor)	Exterior	Interior	Structure
2. Concretized (* ElementModelor)	Interior	Structure	Exterior
3. Detailed (* PartModelor)	Structure	Exterior	Interior

be methods and processes that guide the designer through.

Three design-methods have been identified:

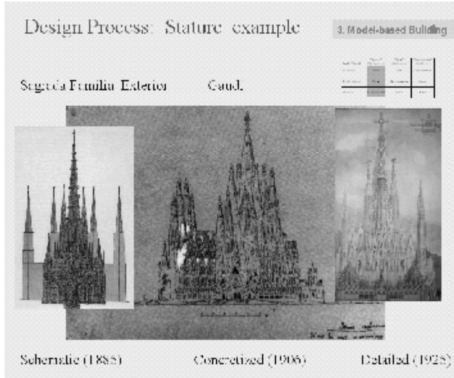
Stature: where the designer starts from outside sculpturing the building mass - the stature.

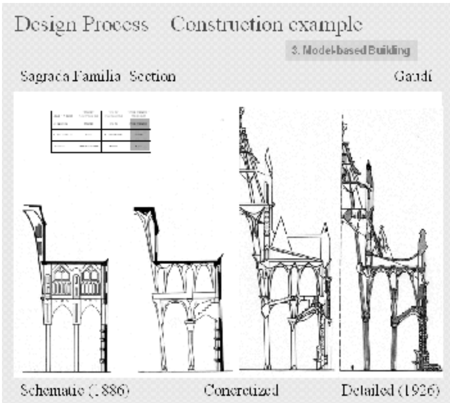
Room: where the designer works from inside, creating and organising the functional spaces.

Construction: where the structure of the building is the outset.

Each design-method goes through the same process:

From ***Schematic*** design via ***Concretised*** to ***Detailed*** design.



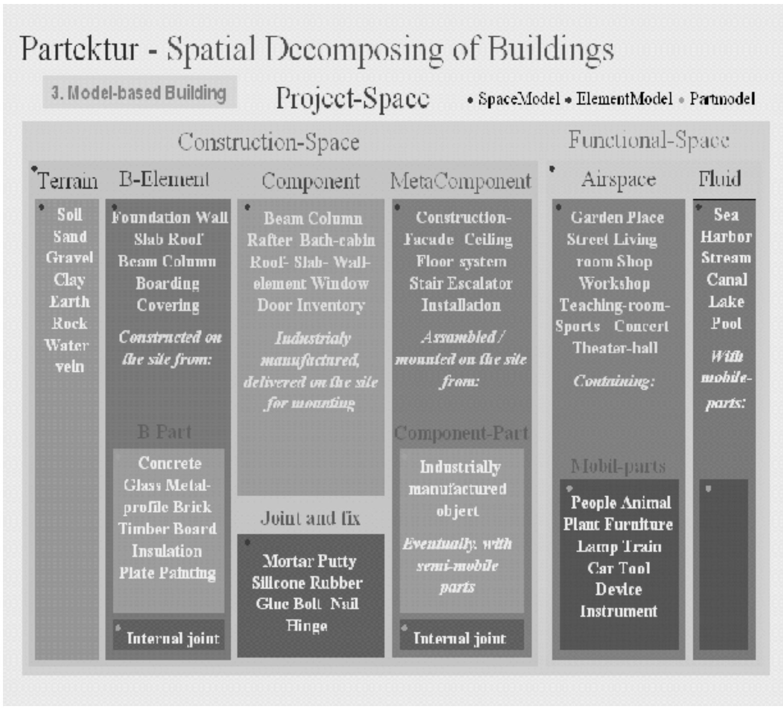


A cursory investigation of among others Antonio Gaudí's work on La Sagrada Familia [1] confirms that these methods and processes are in use, and that the design process tends to start with schematic layout of the functional spaces stated in the room-program, or with the schematic form of the building masses, or with a schematic structural layout, or a combination of the three, and then continues with successive concretizing towards a fully detailed plan. But it also shows that external, internal and constructional methods are mixed, and that the process from schematic to detailed design is not always followed in a strictly linear manner.

This leads to the conclusion that the B-processor should support all three methods: Stature, Room and Construction driven design, so that the methods can be used concurrently and mixed. And that the B-processor must support non-linear design decisions along the line from schematic to detailed design. This makes a demand on the Partecture to be able to hold design-proposals for a given building object on any detail-level, and calls for Modellors to be able to keep design-objects dynamically updated as the design process goes on.

Spatial Building Decomposition

As mentioned the building can be seen as a division of the total ***Project-space*** into spaces containing the building ***constructions*** and spaces containing the ***functions***. This view of the building supports the concept of the design process as a successively more and more precise geometric definition of the surfaces describing the transition from one space to another - floor to room, wall to roof, etc.



The following proposal for a spatial decomposition of the building comprises a set of spatial types in a hierarchical spatial structure with the outer space as the ultimate limit and the Project-space as the actual container of the building.

Project-Space, at the top of the spatial hierarchy, extends in the height from the bearing layer in the terrain to up above in the air, and in the width as long as the building can be seen, heard, smelled, but in practice with the site limit as the interesting line. The Project-Space consists of two types of spaces, filling up the total Project-space - the ***Construction-Space*** and the ***Functional-Space***. Where the Construction-Space contains solid material - the building elements/components - and the Functional-Space is the air/water where the functions - the human activities - take place.

Construction-Space, divided into the spatial types:

Terrain, with spaces for all the different materials in the ground.

Building-Elements, all structures that are constructed on the building site from:

Building-Parts and

Internal-Joints

Components, arriving prefabricated to the site to be mounted in the building

Meta-Components, that are assembled and mounted on the site from

Component-Parts and

Internal-Joints

Joints-Fixes, that bind B-elements, Components and Meta-Components together in the building

The Construction-Space can contain semi-mobile objects, objects that can turn and/or do a linear movement in accordance with some limits.

Functional-Space, divided into the spatial types

Airspace and ***Fluid***, containing:

Mobile-Parts, people and their devices, for the use of the building.

In the process of spatial building decomposition there was no need for more than three hierarchical levels, measured from below the Project-Space, but practical use could show a need for more levels.

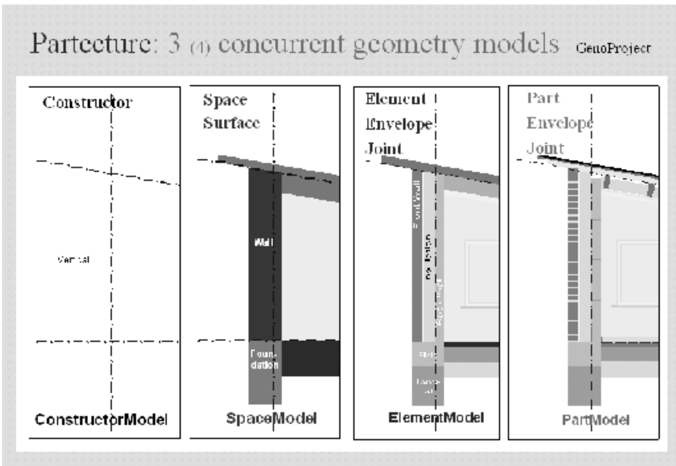
Partecture, designed for dynamics and details

The general demands on the B-processor, the considerations of design-process and the proposal for spatial decomposition have led to - and argue - a new design for the Partecture.

The Partecture [13] is designed to represent spatially the Building objects in three versions –space, element, part - with increasing degree of detailing, like the three hierarchical spatial levels in the decomposition and following the three design-process levels.

The proposal for a new Partecture has on the *Schematic* design-process level a representation for the Functional-Space and the Construction-Spaces called the ***SpaceModel*** [2].

On the *Concretised* level, and contained in the spaces of the SpaceModel, there is a representation of the Construction-Spaces content: Terrain, Building-Elements, Components, Meta-Components, Joints-Fixes, and of the Functional-Spaces content: Airspace and Fluid. This is called the ***Element-Model.***



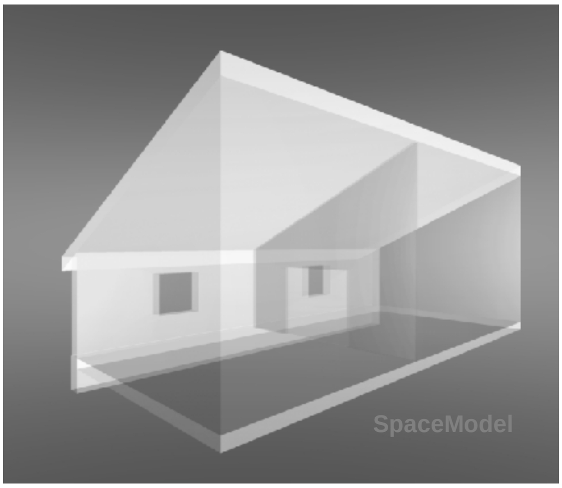
And on the *Detailed* level there is a representation of Building-Parts and Internal-Joints contained in Building Elements, of Component-Parts and Internal-Joints contained in the Meta-components, and Mobil-Parts contained in Airspace or Fluid. This is the ***PartModel***.

Above the SpaceModel-level corresponding to the Project-Space-level there is the ***ConstructorModel***. It is a representation of gridlines, site limits, story height levels, etc. that monitoring the surfaces of the SpaceModel.

ConstructorModel

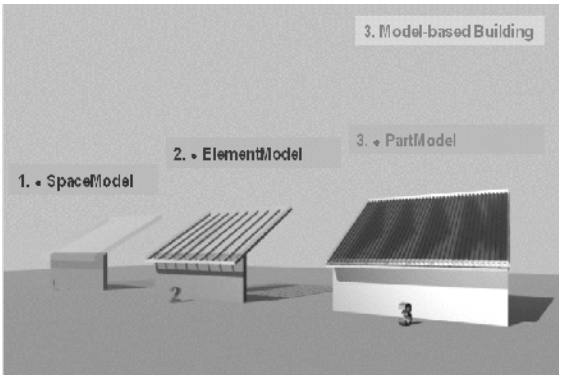
The ConstructorModel is hardly a model, it is more a representation of a geometric constraint - called a ***Constructor-*** that can be stated in the form of a point, line curve, plan, etc. Constructors are related to surfaces in the SpaceModel, and are thus a superior “restrainer” of the design, as it develops

SpaceModel



The SpaceModel spaces - Functional-Spaces and Construction-Spaces - are defined by ***Surfaces.*** Each Surface is a transition between two and only two spaces. That means that the surface defines the geometric form of the spaces and the relation between the two spaces. The SpaceModel is meant to be a sketchy and strongly interrelated model that supports Schematic design.

ElementModel - PartModel



Building-Elements: Components and Meta-Components and Joint-Fixes - at the ElementModel level - are geometrically a subdivision or fill-in, - like a spatial Boolean subtraction- to a space in the SpaceModel. The fill-in can geometrically have its own format like the ArchiCAD GDL.

In the same manner: at the PartModel level Building-Elements are filled-in with Building-Parts, and Internal-Joints, and Meta-Components with Component-Parts and Internal-Joints.

Although the Partecture supports a linear design-process - schematic to detailed - it is possible to model concurrently at all the four model-levels. If the modelling is done top-down a detail-part has a spatial limit at the level above that allows implementation of more or less automatic modelling tools - Modellors - specialised to model a specific Building-Element, Meta-component, etc. into these spatial limits. The strong spatial interconnection between spaces at different model-levels has the advantage that a dynamic automatic updating of Building-Objects - Modellors - is possible in response to changes - size, position, material, etc. - in neighbouring Modellors.

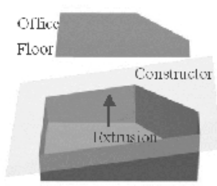
Modellor

As stated above the Modellor is both a specialised modelling tool and the IT-object that represents the Building-object in the Partecture. The intelligent Modeller will support creation of the object and keep it updated during the design. The Modeller will on request from an Evalor deliver relevant data to support analyses of a design aspect. The accuracy of the analyses will follow the model-level. There will be specialised Modellors for each of the model-levels in the Partecture model hierarchy.

Constructor-, Space-, Surface-Modellor

The ***Constructor-Modellor*** sets up a Constructor: all kinds of lines, curves, plans, etc. representing site limits, storey heights, roof angle, grid lines, etc. The surfaces of the SpaceModel can be related to - parallel with, a distance from, etc - a Constructor, and is following the Constructor if it is moved. So the Constructor forms a superior regulation mechanism of the design.

The ***Space-Modellor*** is the tool that has the role of defining spaces, the name, the type: wall, kitchen, stair, garden, etc., and as spaces can be defined in the model before they are given geometrical attributes at all, as a place in the Partecture to note ideas - max/min size, description of content - about the new building in the



earliest design phase. A ***Surface-Modellor*** gives the SpaceModel geometrical life by defining the surface and relating it to its two spaces [2, 13, 22].

Construction Modellors

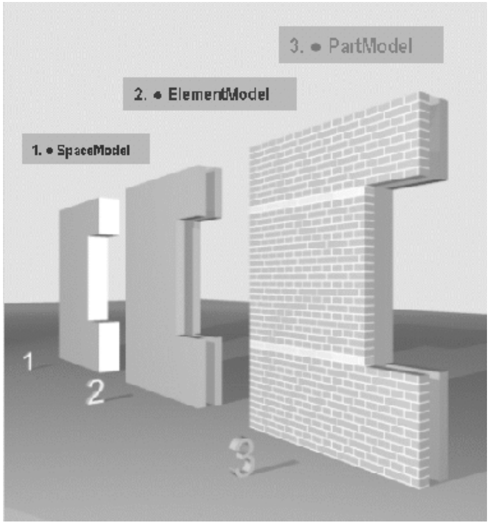
The ***Element-Modellor and Part-Modellor*** are in the general form tools to model a Building-Element - B-Element-, a Component or a Meta-Component, and also Modellors to detail with Building-Parts or Component-Parts.

The Modellor is specialised to a specific Building Element – a Wall-Modellor, Roof-Modellor, etc. - or specific Building-Parts –Tile-Modellor, Brick-Modellor, etc.. Or it is as the Meta-Component- and Component-Modellor product-specific – a Velux-Modellor (roof light), Ifö-Modellor (sanitary), Vitral-Modellor (glassing), etc.-.

Building Element Modellor

All buildings, even the most industrialised, contain constructions - B-Elements - that are made on the site. The Building-Element is constructed on the site from Building-Parts, and is modelled by specified Modellors.

A Reinforced Concrete-Modellor is specialised to design the mould and the reinforcement. A Boarding-Modellor is specialised to support different layouts - two on one, ship-lab, etc.



The general B-Element modelling process is:

1. A Space-Modellor models a Construction-Space for the B-Element in the SpaceModel
2. An Element-Modellor subdivide in spaces for B-Parts in the ElementModel
3. Part-Modellor models Building-Parts in the PartModel

The specific process for an external wall is:

1. A Wall-Space Modellor creates space for the wall
2. A Cavity-Wall Modellor creates spaces for the external and inner wall.
3. A Brick-Modellor models the bond of the facing wall. A Wall Insulation-Modellor models the insulation, and a Wall-Block-Modellor - Leca-Modellor - creates the inner wall.

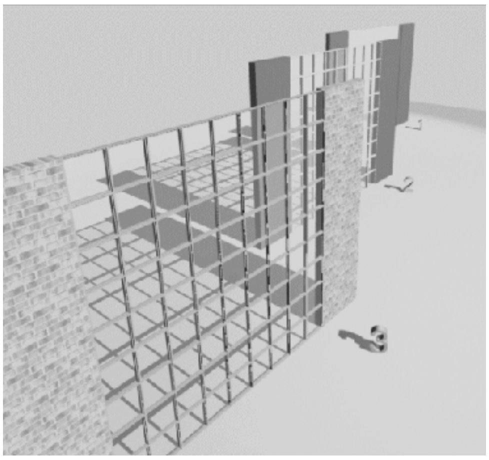
The interesting thing here is that the Modellors work inside geometric limits set at the model level above, and as mentioned this enables the development of parameterized, specialized - with knowledge of the nature of the construction - and automatic Modellors and that the Modellor can make intelligent up-dating of the element it represents. In this example it adjusts to detailing in the wall-top and -bottom and to holes for windows and doors. Thus the proposed Partecture and Modellor concept will allow for a very dynamic design-process. As will be shown in the following, it is not always necessary to go through all three design steps. Step 1. could be omitted and step 3. could be replaced by a specification.

Component- and Meta-Component Modellors

Technically speaking the Component- and Meta-Component Modellors will often be a plug-in/add-on to the B-Processor, and it will be obtained from the component manufacturer as part of their product information.

Components are delivered on the site in one piece and mounted into the building. A Component-Modellor, a special case of the general Element-Modellor, and works in it simplest form like this: The product-specific Component-Modellor is found in a www library and down-loaded. During the insertion in the model, parameters specifying the component are set and the integration in the Partecture is done automatically: the Component-Modellor gets its own spatial place in the SpaceModel, and so it can act intelligently and dynamically.

Meta-Components come to the site from the manufacturer as Component-Parts, and are assembled and mounted into the building without any further shaping of the parts.



The general design process with Meta-Components is similar to that of Building-Element design. In a concrete construction of for example a glass façade the design process goes like this:

1. A Glass-Façade Space-Modellor models the Construction-Space to contain the glass façade.
2. A Glass-Façade Modellor, divides the Construction-Space into Element-Spaces for frames and glass and supports experimenting with proportions.
3. A product-specific Part-Modellor, for instance a Vitral-Modellor, models the full detailing.

Step 1. where the façade is represented by a transparent space, is convenient early in the design-process but could be omitted. The same is valid for step 2., if the proportions are settled.

The B-Processor will allow direct modelling in the PartModel and will automatically define the connections to the ElementModel and the SpaceModel.

As can be seen in this example the Modellor can have design supporting tools - here proportioning - built in. Another important aspect is that the product specific Modellor will be able to produce manufacturing data to be used directly in the production plant, making a direct line possible from design to production, and also the possibility of designing with all the potential component variants offered by the production line.

Model-based Building. Scenario

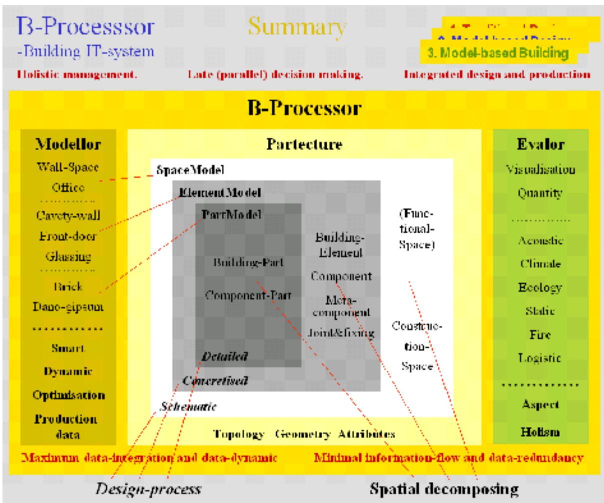
The implementation of the B-processor - Model-based Construction - demands and makes possible support, organizational changes and thereby enhanced processes.

The architect will be able to conduct a central role in close collaboration with the client, and in partnership with the engineer and contractor as a holistically oriented responsible leader of the design and production of the building.

Architects and engineers will as building-component designers move into a new area as with each new component they have to develop a corresponding Modellor to be down-loaded by the building designer to replace the traditional product-information.

There will be a new area for engineers and other experts in developing Evalors, with expert knowledge incorporated.

The dynamic B-Processor supports a decision process, where the final decisions for a design section can be delayed until the moment when the context with the surrounding sections is clarified as well as possible.



The holistic attitude to the building-process is supported. Product data and handbook knowledge are built into Modellors and Evalors and thereby made directly accessible to the designer without a middleman. This is valid for well known constructions, whereas with new construction development collaboration between architects, engineers and contractors is supported by the Partecture as a central design-medium.

The Partecture also plays a central role during the construction of the building as an integrator and source of production-data. The legal role of the traditional design documents will not be played by the Partecture in a partnering organization, but the B-Processor makes a more prototype-like design-method possible, where not everything has to be fully detailed before construction begins.

In the planning of the production process the Partecture will be used to model scaffoldings, moulds, cranes, all of which need some space available. Also the schedule will be illustrated as an animation of the building process.

The success of implementation of Model-based Building depends upon many things, but especially three are of vast importance:

- A change of attitude among architects.
- A contemporary shift to modelling-as a minimum in the actual building project team.
- A redesign of the B-processor to make it efficient and dynamic enough for Model-based Building.

Literature

1. Cesar Martinell y Brunet, Gaudi –his life, his theory, his work, Barcelona, (1967)
2. Kristian Agger, Datastructures for function and geometrical description of buildings, paper, CAD76, London, (mar. 1976).
3. Bryan Lawson, How Designers Think, Butterworth Architecture, (1980).
4. Ulrik Løssing, Kristian Agger, Ejgil Nybo. Architectural Design With Computers -a pilot project 87- ISBN 982875-3-2, Aarhus, (1989)
5. Tom Søgaard Larsen, Projekt 90, rapport, PAR, København (1990).
6. Laila Dybkjær, Computer-Aided Floor Plan Sketching –A study of Knowledge Representation in Architectural Design, Ph.D. Thesis, DIKU, København, (1991).
7. Jan Karlshøj, Principper og metoder for opstilling af datamodeller til byggeteknisk anvendelse, rapport, DTU ABK, (1994).
8. Per Galle, Product Modelling for Building Design, - annotated bibliography, DTU, Lyngby, (1994).
9. Bo-Christer Björk, Requirements and information structures for building product data models, VTT, Espoo, (1995).
10. Carsten Rode og Karl Grau, Implementation of a pragmatic system for integrated Building analyses, SBI, Hørsholm, (1996)
11. Kristian Agger, Traditionel projektering eller projektering med IT, AbbNYT nr. 6 Taastrup (1997)
12. Kristian Agger, IT modellering, - Status og Perspektiver, (Abb)NYT nr 1, Taastrup, (1998)
13. Kristian Agger, Geno-Object-Classes in Construction IT, CIB W78, KTH, Stockholm, (1998)
14. ArchiCAD 7.0 GDL and Reference Manual, Graphisoft, Hungary, (2001)
15. Industry Foundation Classes, Release 2 (and 2x), IAI Members CD, IAI, (2001).
16. Projekt Hus. Temagr. 7. Forslag til Udviklingsprojekter, Peder Gammel, Kristian Agger og Hans Peter Svendler Nielsen, CID (2000)
17. Byggeriets Fremtid – fra tradition til innovation, Byggeriets Task Force, By- og Boligministeriet og Erhvervsministeriet, (2000)
18. Activity add-on, ArchiCAD, Anders Ekholm, LTH, Lund, (2000)
19. Ny organisation i byggeriet - et bygherrekrav. Kristian Kreiner. Handelshøjskolen i København. Byggeriet *Net*-op nu/Notat CID (2001) and Organisering af udviklingsprojekter, Teknik og Data (2002) http://www.frinet.dk/index_2.asp
20. Using spaces with IFC R2. Jiri Hietanen / BLIS –project, Helsinki(2001)
21. The Essential Gaudi. Jordi Bonet i Armengol (English Mark Burry) Barcelone (2000)
22. The Future of Interoperability and Use of Product Models. Summary of a Delphi Study (a working paper, not for publication!). Tapio Koivu and Hans Björnsson. Stanford, October 4, 2001
23. SketchUp 3D (surface-modellor): www.sketchup.com (2002)
24. IFC pilot project. IHK, Copenhagen (2002). www.IAIforum.dk