PRODUCT DATA MODEL FOR INTEROPERABILITY IN AN DISTRIBUTED ENVIRONMENT

RICHARD JUNGE
Faculty for Architecture, Professorship for CAAD
Technical University Munich

DR. THOMAS LIEBICH
CAB Research & Consult
Munich

Abstract

This paper belongs to a suite of three interrelated papers. The two others are 'The VEGA Platform' (Junge, Koethe, Schulz, Zarli, Bakkeren. this volume) and 'A Dynamic Product Model' (Junge, Steinmann, Beetz. this volume) These two companion papers are also based on the VEGA project. The ESPRIT project VEGA (Virtual Enterprises using Groupware tools and distributed Architectures) has the objective to develop IT solutions enabling virtual enterprises, especially in the domain of architectural design and building engineering. VEGA shall give answers to many questions of: what is needed for enabling such virtual enterprise from tht IT side. The questions range from technologies for networks, communication between distributed applications, control, management of information flow to implementation and model architectures to allow distribution of information in the virtual enterprises.

This paper is focused on the product model aspect of VEGA. So far modeling experts have followed a more or less centralized architecture (central or central with 'satellites'). Is this also the architecture for the envisaged goal? What is the architecture for such a distributed model following the paradigm of modeling the 'natural human' way of doing business? What is the architecture enabling most effective the filtering and translation in the communication process. Today there is some experience with 'bulk data' of the document exchange type. What is with incremental information (not data) exchange? Incremental on demand only the really needed information not a whole document.

The paper is structured into three parts. First there is description of the modeling history or background. the second a vision of interoperability in an distributed environment from the users coming from architectural design and building engineering view point. Third is a description of work undertaken by the authors in previous

project forming the direct basis for the VEGA model. Finally a short description of the VEGA project, especially the VEGA model architecture.

Introduction

Having a close look at the daily work of architects, it becomes obvious that a very considerable portion of their time and efforts is spent with organizing the collaborative design process and communication with the client, official bodies construction companies and other engineers. The pressure coming from these duties is rising, and special project managers are employed not only for large projects. By the same time the responsibility and influence of the architect on the whole building process, which does not end with the completed design drawings, is reduced, part of their traditional scope of the process will be lost.

Considering the roots of the profession an architect was the „master of builders“, as the Greek origin of the name shows. There were no special means necessary to allow him to communicate with his fellows, all were on site and design specifications where often drawn directly on stone or earth. Sketches merely served as schematic overviews. The first change of this paradigm appeared during the quattrocento, when the Renaissance architects introduced new design methods and were separated from the building site. Detailed and scaled design drawings now became part of the contract and had to be communicated with the builders. With the increasing complexity of buildings, and the parallel process of further division in work, far more detailed design specifications have to be communicated and coordinated with a growing number of participants. The question here is, whether the developing technology can provide the appropriate means to allow architects to maintain their central role or whether they will become design professionals amongst others being coordinated by project managers working directly for the client or main contractor.

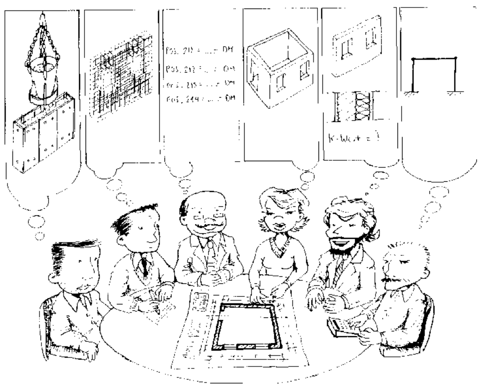


Figure 1 Collaboration today

On the technological side of the problem there is the overall impression that efforts where mostly assigned to the development of individual software as for drawing and designing, both within commercial sector and research while the communication aspect was almost neglected. At least the communication aspect was understood as an integral part of architects professional work In consequence the often quoted „islands of information“, or islands of construction computing, have to be acknowledged. How to bridge those and to allow design programs to collaborate on behalf of the designers as natural as designers do? What could be expected from such a technology and what are the main components?

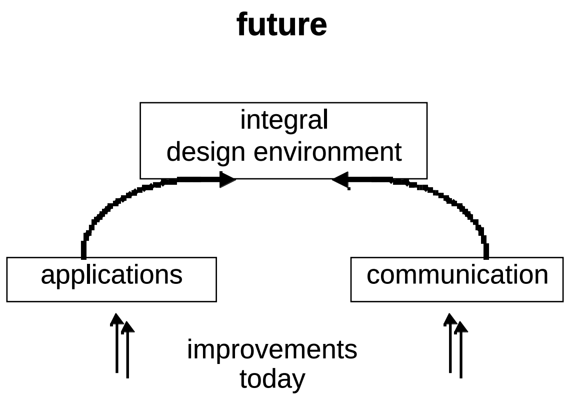


Figure 2 Improvement of IT

Basic Technology

Such a ‘future integral design environment’ modeled after the human reality of design and communication needs at least two basic technologies. First software has to understand the semantic of the objects it is dealing with. The solution is a ‘semantical kernel’ as a basis for applications, a product data model. The second is a ‘transport system’ for information, not just data, that allows direct and flexible communication between, again following the human reality, distributed applications and their product data models. The technology for this part is COAST (COrba Access to Step models) which is an extension of CORBA (Common Object Request Broker Architecture). This paper will focus on the product data model part, COAST and CORBA technology is being explained in ‘The Vega Platform’ (this volume).

Product Modeling is nowadays a widely accepted methodology for the development of advanced software systems in engineering domains, such as architecture and building. The methodology improved a lot especially during recent years through European projects such as COMBINE, Neutrabas, ATLAS; COMBI; PISA and since 96, VEGA as well as a lot more from other then the AEC domain. The progress however is not based on the fact that these projects are all EU funded but rather on the fact that all these projects where following the basic idea: using and at the same time improving the technologies and methodologies of the STEP environment. As a result there is nowadays a powerful basic technology of STEP- CASE- tools available as well as a widely accepted baseline towards product modeling.

The modeling approach or architecture in these projects was different partly, certainly, because projects had different goals. These developed from integration of applications in specialized domains to the development of a platform for interoperability through distributed product models. Another reason for different model approaches certainly was the state of knowledge at the corresponding time. The very origins of product models are to be found back at a time when the term product model wasn't even invented.

A starting point surely is the generation of systems like GLIDE and BDS (Richens, 1978) and many others. The next period is defined by a first generation of expert systems (Landsdown, 1983), (Wager, 1984). Both approaches are using data models. Today one would say they where using Building Product Models, or to follow B-C Bjoerk's more precise distinction, Building Product Data Models (Bjoerk, 1995).

A certain degree of consensus, e.g., that an integrated building design system has to be structured around building parts and not around geometrical objects was already reached in the early years (Eastman, 1978). Nevertheless it took more then a dozen further years until the structures of building product models have become clearer (Junge, 1994).

The stream to which above mentioned projects belong to started with the development of two generic models. The GARM (General AEC Reference Model) (Gielingh, 1988) was developed for the STEP project (Standard for the Exchange of Product model data). The GARM provides a first basic concept by introducing the 'Product Definition Unit' (PDU) and its subtypes 'functional unit' and 'technical solution'. The Building Systems Model (Turner, 1990), developed as an input to STEP, also shows a top down strategy to model a building. It therefor introduces functional systems as, e.g., enclosure, structural, mechanical, etc. and their entities.

With the development in the IMPACT project very intensively studied fundamental modeling principles as specialization, discrimination and orthogonalisation as well as implementation principles as extension and instantiation. The model introduced the technique of 'layered models (IMPACT 1991). This term was used for a concept of a structure with a number of models of different levels of abstraction on different 'layers within one larger model structure. This as a significant architecture for the design of models with a larger scope.

The ESPRIT project Neutrabas (Neutrabas, 1991), which is in the ship building domain, used concepts developed in IMPACT at the same time and extended product modeling technique further. It is one of those projects that started to deal with the question, respectively solutions, for objects, belonging to more than one system and are playing different roles in them. The discussion on this problem is still open today.

This open discussion leads directly to the question how a building product model that would cover the needs of all parties involved in the process of design, construction, maintenance and management has to be structured. Today the belief is that there is no way leading to a homogeneous single central building product model is accepted nearly unanimously. In a number of current projects one finds various terms used for strategies dividing the universe of discourse into partial models. Terms used to describe such partial models are 'topical model' (van Nederveen and Tolman, 1992). The ESPRIT project ATLAS has 'view type models' (ATLAS, 1994) and ESPRIT project COMBI uses 'partial models' and 'application models' (Junge et al, 1994). Although the solutions are slightly different, a way has been found that leads to practical building product data models. Currently, the discussion focuses on the question, how to interconnect these partial models?

The discussion is on strategies where these partial models should be compatible with each other. The idea is that there is a "central" part of the modeling domain that would be shared by all partial models (Luiten et al., 1991). These central parts are often called kernel or core models. Again one finds this concept in ATLAS and COMBI, as LSE Project Type Model or as Central Neutral Model The main question still remains: What is the focus of a core model? Is it the sum of parts used in at least one application or domain? Or should it be a small but generic kind of ‘overhead model` which aims at staying stable when the scope of the overall model expands? A proposed solution for the first is e.g. the minimal kernel model as used in the NICK project (Tarandi, V. 1993)

The Building Construction Core Model (BCCM, ISO STEP Part 106) (Tolman and Wix, 1995) is based on the opposite concept. Here the basic assumption is that all objects that are used by more than one system should go into the core model. This at first glance simple concept implies that the core model will explode into a kind of central model. The question is how many objects do not belong to more than one system? This seems to be a concept that bears the danger in it that the model will ‘explode’ because of the shear number of entities it will have to deal with and it will not be a stable core for the suite of models it is intended to serve until all these models are being developed themselves.

It seems that the BCCM is following a certain ‘One Model Mystique’, everything that is used in more than two apps has to go into the model. Such a model for Computer Integrated Construction (CIC) could easily reach a number of some thousand entities. Even if most of the attributes would go into the connected ‘domain models’, thus leaving itself to be a semantically flat model, it seems very questionable if such a model would be manageable. A comparable approach, the IPIM, was already discussed in STEP during '88 and '89 and it was given up.

Period	Focus of development	Example Model
1975-85	'pre-product models'	
1975-80	basics of building models	GLIDE
1980-85	models for expert systems	
1985-	building product data models	
1986-90	generic product models	GARM, BSYSM
1990-95	aspect and layered models	COMBI, ATLAS
1995	'consolidated models'	O.P.E.N

Figure 3 Model history

One could draw a kind of time table showing the main focuses of research and development of product modeling technology. A variation of such a time table presented by Bjoerk (Bjoerk 1993) is shown in fig 3. The concept of 'consolidated models' is introduced into the discussion in this paper for the first time.

Model layer	Example data structures	Example models
Fundamental data model	objects, attributes, relationships, generalisation- specialisation, rules, methods, messages	Relational data model Entity-relationship -model The object as in objectoriented programming The frame
Generic product description model	Data structures for describing aggregation- decomposition, type objects, versions, shape and location information, abstraction levels etc	STEP Resource Parts EDM GARM OOCAD
Building product data model kernel	Generic object classes which are particular for a specific design or construction discipline and or phase in the construction process	Building Systems Model GSD Model RATAS Framework KBS MODEL NICK
Aspect models	Detailed object classes which are particular for a specific design or construction discipline and or phase in the construction process	COMBINE IDM RATAS quantity take off prototype Structural Steel Model CIMsteel LPM
Aplication models	Detailed object classes which constitute the conceptional model of one particular application	Not subject for this framework

Figure 4 The layered Framework Architecture

In 1994 Bjoerk presented a structure of how the different product modeling concepts very briefly and incomplete described above can be grouped. He called it a ‘Layered Framework Architecture for grouping data structures which are used in building product data models’ (Bjoerk 1994). Bjoerk writes: ‘The set of data structures that make up a fictive comprehensive building product model is partitioned using an onion- like overall architecture. ....a decomposition into five layers would seem appropriate’.

Following Bjoerk all these layers combined would form a comprehensive building data model. The models quoted in the framework where more or less focused on the necessary exploration of only one the layers. From today’s level of knowledge it is obvious that none of those ‘layers’ as standalone solution can provide the necessary basis for a comprehensive Building Product Data Model.

The authors therefore propose the ‘Consolidated Model’ as the attempt to combine the pieces developed in previous projects to an architecture mature for an commercial implementation of the paradigm of the ‘Building Product Model’.

Steps towards the ‘Consolidated’ Model

The Consolidated Model architecture emerged out of a more or less consequently and gradually development in a series of projects in which the authors where involved during the last five years. These are COMBI, an ESPRIT project, NextCAAD I, NextCAAD II, both are industry funded development projects and VEGA, again an ESPRIT project. One common basis for all these projects was an as much as possible and consequent application of so called ‘STEP methodologies’ and the search for a model architecture flexible and comprehensive enough to serve as a basis for commercial applications in architecture and engineering.

COMBI

The COMBI project (1993-95) developed a prototype environment for cooperative design focused on the domain of structural engineering. It envisions an intelligent environment based on the idea of ‘integration by communication’ (Junge 1991). The first focus is the development of four application tools for structural design. These tools form a chain beginning with beginning with soil mechanics and foundation design to structural analysis end ending with reinforcement design. The second focus is on integrating these four application tools, while the third focus is on linking tools which are developed on the basis of a product model with an external traditional CAD system. For these three tasks a product model framework is being developed.

This framework has been created under consideration of the evolutionary multi stage and multi agent nature of the design process and for unified integration approach. The architecture of the framework can schematically be represented by four hierarchically structured levels (Figure 5). It can be envisioned as a network of computers and users, where communication and coordination is achieved through a shared information medium and a control mechanism which provides facilities for the integration of loosely coupled design tools.

The application tools, that support the work of the individual designers shown on the bottom level have their own application dependent product model, which need not necessarily conform to a standardized product model specification. Thus, application data are processed and stored only within their application domain, which helps to avoid information explosion and unnecessarily complicated data structures. The three main reasons for this decision had been:

- the application data structure must be organized according to the needs of the application methods in order to achieve maximal run time performance,
- existing tools must be integrated without internal modifications, and
- specific data extensions needed by the application methods are often only for temporary use and can be generated automatically, e.g. by a finite element mesh generator, as it is the case in the COMBI prototype.

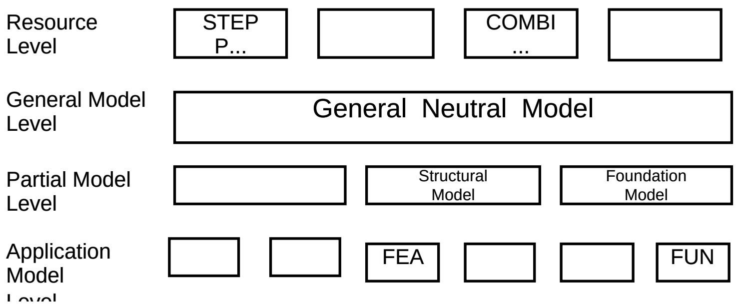
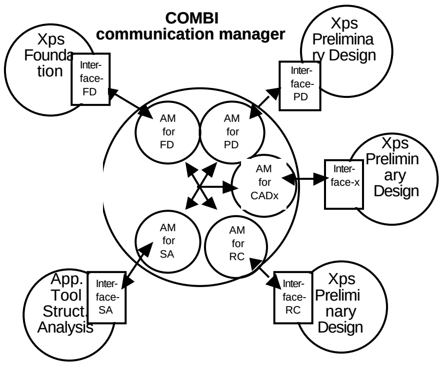


Figure 5. COMBI Framework

The two middle levels are forming the common kernel of the integration framework. They are divided into a central neutral model and several COMBI partial models designed to serve specific engineering domains. Communication within one domain is supported by the corresponding partial model, communication between different domains need additionally to their partial model the Central model and thus increasing the need as well as the complexity of the necessary mapping mechanisms. The uppermost level serves as a common source, containing resource parts from STEP as well as small COMBI proprietary resources. The framework can be classified as a layered architecture using a central kernel and partial models.

COMBI's application models contain all possible output/ input entities and their attributes from the corresponding application tools. With the help of an intelligent center, a tool capable of filtering and translating the incoming information for use in the receiving system, however, the desired communication can be achieved in a more flexible way. The model translation/ conversion technique used by the COMBI Communication Manager (CCM) knows the origin of the entities respectively their attributes within the sending systems. Therefore the CCM performs the translation between different specification forms of design objects and attributes within the participating systems. The rules that describe the input and output data format and the transactions are realized and executed in KEE. The CCM establishes the exchange of instances between files using STEP physical file format between different models.

Figure 6 COMBI’s CCM



NextCAAD I

The objective of the project is to develop a CAD system that is not primarily oriented towards the graphical and geometrical processing as current systems are. the basic idea is that the architectural design object is considered as an logical entity within the product model. Architectural design objects can be rooms, spaces, building components of different complexity, grids, etc.. Parts of the objects’ definitions are dealing with its physical shape, physical properties, others with its representation, which is always an abstraction of the real design object.

The existence of an object is not based on the representation of its geometry on the screen respectively the object's definition in geometry data. This is the fundamental step, which creates the basis for a consistent application of the product model philosophy. An object is a more abstract entity that can be viewed from different viewpoints. Thus it has different representations, e.g. different geometric representation, characterized by different sets of attributes.

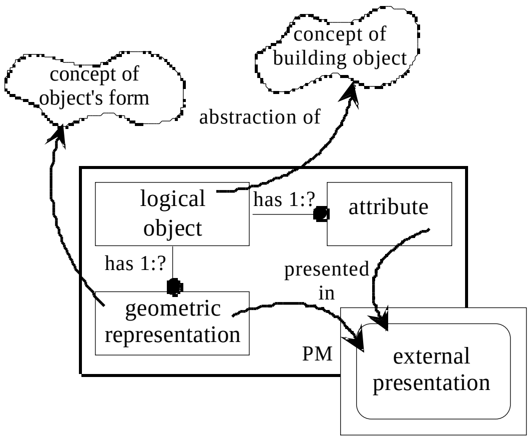


Figure 7 The logical object as basic entity

An application covering a whole area, such as CAAD the discipline of architecture, has to be dynamic by its own nature. It should be possible to shift the focus from, e.g. the spatial layout, to the building components and again to the bill of quantities.

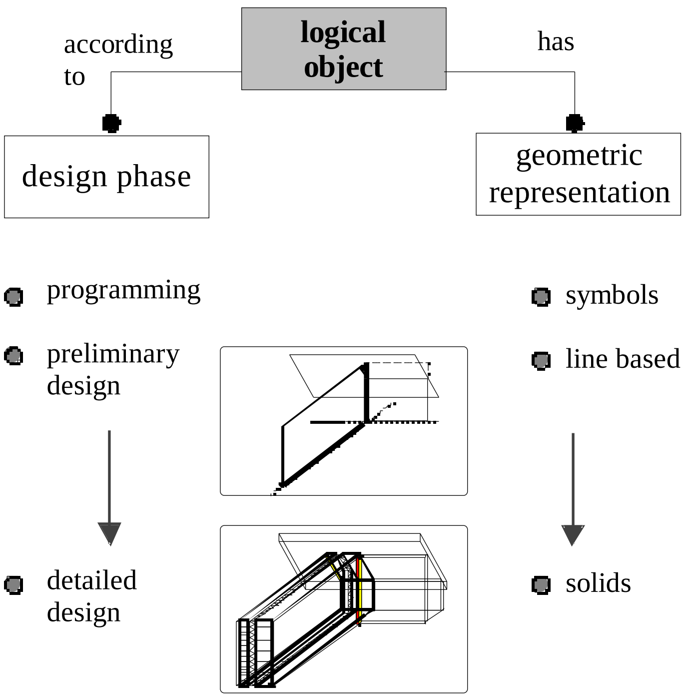


Figure 7 Different geometric representations of an design object

Therefore the modules of the system, and accordingly the partial models of the product model, should be interconnectable in a dynamic way. Thus, the underlying product model is structured as a web of interconnected modules, in opposite to the layered approach, e.g., in STEP.

The next generation CAAD has to deal with many phases of an object's life cycle, from programming phase to detailed design. Objects may occur and disappear during the life cycle, usually they are transformed or refined many times. In particular the geometric representation change from vague and fuzzy symbolic descriptions to sketch-like two dimensional geometry and again to exact two- and three dimensional geometric entities. Thus, multiple geometric representation is of special concern for the project.

Design is a highly dynamic process with many diverse particular activities and constantly shifting focuses between them. Translated into a product model this means that there are many individual focuses. This could be modeled as a huge singular model. Disadvantages are discussed in the introduction. Even if it where theoretically possible to model such a thing, practically it is not feasible, because one has to have particular results in a foreseeable range of time. All this leads to modular models. The conceptual idea for bringing the dynamic nature of the design process into a product model was that of a number of modules that could be connected together in dependence of the momentary context requirements of the CAAD system. This is the idea of a web of dynamically interconnected modules.

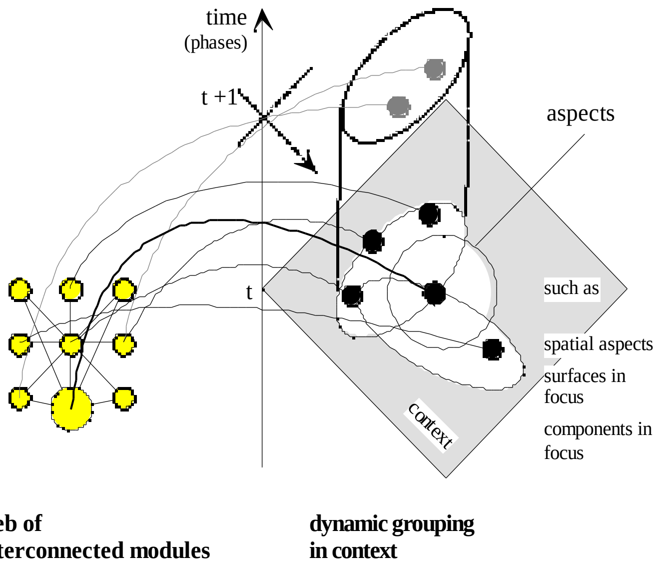


Fig. 8 dynamic grouping of interconnected modules

The concept of this model architecture could not be realized for many reasons. The resources of the NextCAAD I project where reduced and it was focused to architectural design only. This goal could be reached with much simpler model architectures that or based on more proven ground. Secondly it still is doubtful if it would have been possible to implement such an architecture with the implementation technique of that time and project. Today with developments from the VEGA project a realization would be more likely

THE NextCAAD II PROJECT

The model architecture of the NextCAAD II project in a certain way is a step back. However the model is built upon experiences from the COMBI project as well as on

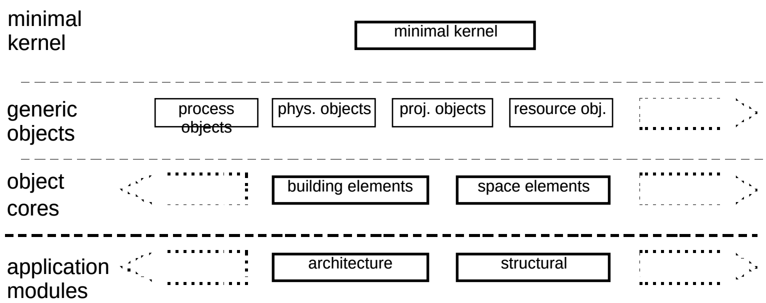


Figure 9 NextCAAD II, modules linked by a minimal kernel

concepts realized in the ATLAS project. Parts that in Combi constitute the ‘general model’ are now splitted into two levels and they are extended at the same time. A small bur generic part is the minimal kernel. it consist out of definitions of generic objects, attributes, association, etc.. The new level, the generic objects level specializes and extends definitions of the kernel in a way that the model finally could be used as an integrated project model as it was the aim of IRMA and others attempted for example. The partial model level from COMBI in way can be compared with the object_core level. The prototype implementation is a slice through the model form the minimal kernel to physical objects and to building_elements down to an architecture application module. The prototype implementation of the model is quite promising.

Figure 10 is showing a small view on the minimal kernel of NextCAAD II. The design object can be regarded as the root object of the model.

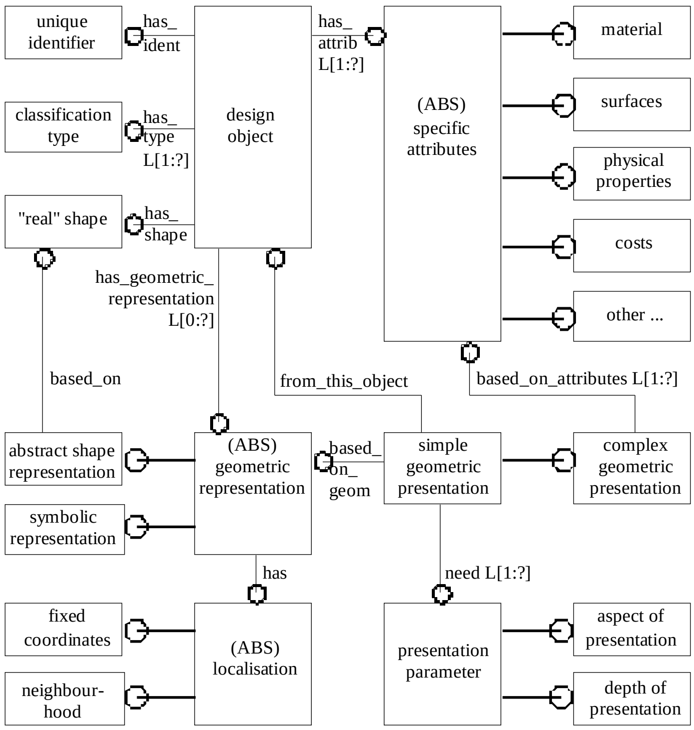


Figure 10 Design object as root of the minimal kernel

The ESPRIT project VEGA

The author is convinced that the only computer based environments really excepted and freely integrated in daily working habits will be those that are working in a similar manner as the human experts of a design and/ or construction team do. Such a team consists of various experts from different domains and expertise. They are highly dependent on each others working results as basis for own work. A strong interactive communication using all kinds of documents and the spoken word is a necessity. The accessibility of information as basis of each others work is based on interpretation which transforms the received information into the experts domain world. The experts are working on the same building elements but having very distinct understandings, semantics of those elements.

One could say all those experts are having different ‘Brain Implemented Models’ (BIM) compared with the models we are used to talk about. There is no person existing with an above all central knowledge, each is understanding their own domain often (or normally?) using different ‘expert languages’. See example of architect and structural engineer talking about a concrete column or load bearing wall. In addition to their own

BIM they have to have a filtering and translation knowledge enabling them to find the mutual implications of a certain decision on their tasks. Without such ‘mechanisms’ it would be necessary for each and every time to communicate in a lecture style manner, surely not a useful type of communication between experts during a design task.

Transferred to an research issue in the IT world this picture of the design/ construction team leads to topics that are very actual today: IT for virtual enterprises, CSCW (Computer Supported Collaborative Work) and distribution of objects ore data models in an network environment to name only a few. Solutions enabling virtual enterprises collaboration in an distributed product data model environment are the objective of the ESPRIT project VEGA (Virtual Enterprises using Groupware tools and distributed Architectures). The task structure of VEGA gives an answer to the question which ingredients are needed for an implementation of the ‘BIM’ idea in an computer environment.

The VEGA project can be structured into five main technical tasks. These are:

1. Conceptual model architectures and product models for a distributed environment (PDM)
2. Technologies enabling communication between applications in an distributed model environment (COAST)
3. Distributed information services (DIS)
4. Work- and information flow management and control (WFL)
5. Implementation architectures and techniques

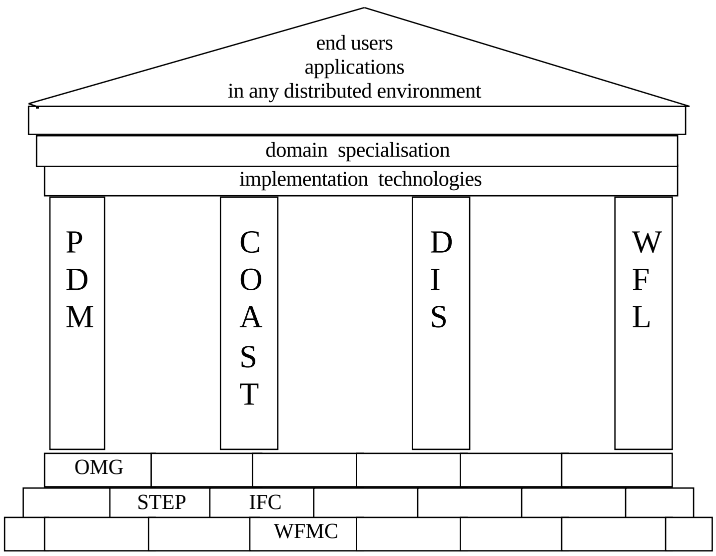


Figure 11 The structure of the VEGA project

To use a picture the first four main tasks are forming columns standing on a basement and bearing an architrave and a gable. The basement bearing the columns are standards like ISO's STEP, OMG's CORBA, WFMC's Workflow interface definitions, IAI's IFC and the Internet with HTML and VRML. And it is not only that these standards or supporting these tasks, there is also a counter reaction back into the standards. OMG has accepted a proposal for extensions of CORBA coming out of VEGA's COAST, there are influences into the workflow interface definitions as well as into IFC's architecture. 'The VEGA Platform' (this volume) is a more detailed description of COAST, DIS and WFL.

The architrave is formed by implementation technologies and domain specialization. These are consisting of two parts: first a generic implementation environment for EXPRESS product data models schemata, the Dynamic Product Model. This PDM among other features allows dynamic evolution of a schema under runtime of the application. The second is a generic product data model which is the necessary task for a specialization of the PDM for the domain of architecture and engineering. 'A Dynamic Product Model' (this volume) is a detailed description of that task. The following is dealing with the 'VEGA model'.

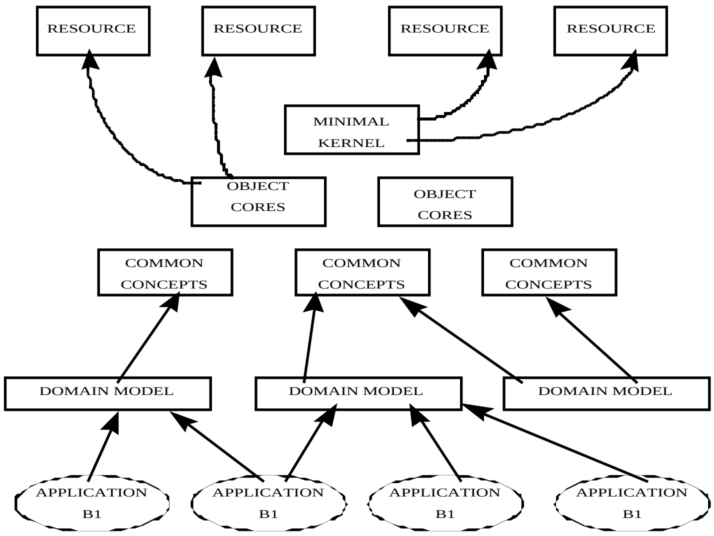


Figure 12 The VEGA model architecture

The architecture of the VEGA model is building again upon COMBI and especially extending NextCAAD II model prototype capabilities. As in the COMBI Framework 'recourses' are used. These can be 'borrowed' and/ or adapted ones, for example from STEP, or native ones, developed for the project. The idea simple is that parts used in several places of the schema should be referenced and not copied into the parts.

The minimal kernel is quite identical with its forerunner, but off course results from prototype implementation are introduced. The two levels which where named generic objects and object cores are now being coupled together under ‘object cores’. It is always a little bit fuzzy to distinct one level from the other or to clearly identify the need for a ‘level’ to be introduced. So one could discuss this separation or not separating of the two. The objective however is similar as before: to extend and specialize definitions of the minimal kernel, which are held very generic into the direction of domain model objects.

A level introduced in the VEGA model architecture is that of ‘common concepts’. The idea behind this is, using the example of a wall for explanation, is the following: In most building product definitions of building elements like wall are being found at the level of aspect or domain models. This results in the fact that there are different wall definitions with respect to their domain requirement or aspect. The level atop those domain models end with definitions of the general building element of which wall is a

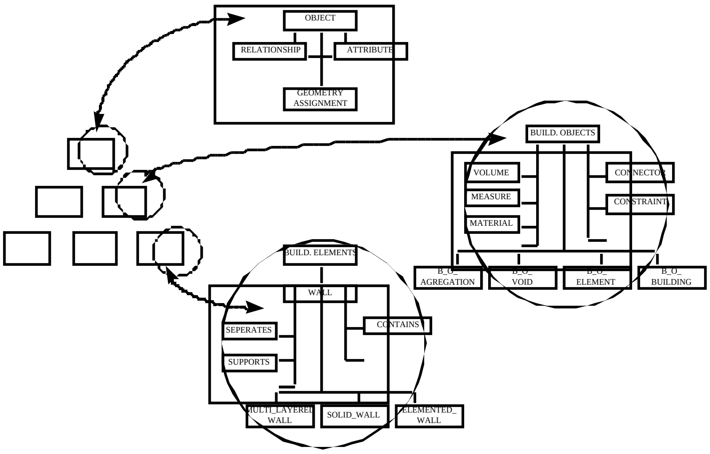


Figure 13. Components of the VEGA Kernel

subtype. The fact that walls are to be defined on domain model level only results in what can be called horizontal mapping or better translations between the different ‘walls’, e.g. the ‘architects wall’ and the ‘structural engineers wall’. These mappings are much more pain making tasks compared with the horizontal mappings at the transitions between one model level to the next lower or higher one. The goal of the ‘common concepts’ is to avoid these horizontal translations and replace them with vertical mappings, in the case of wall from either ‘architects wall’ or ‘structural engineers wall’ to the common concept of wall.

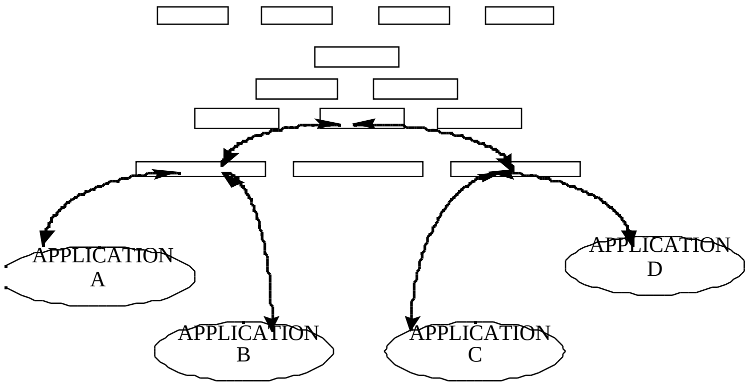


Figure 14 Communication through the VEGA Model

Figure 14 is showing how applications of the same domain (A with B or respectively C with D) are communicating through their domain models. While for communication of A with D, applications of different domains, both using their domain model plus the next higher level which is the common concepts.

By the time of ‘CAAD futures 97’ VEGA will be at its ‘mid-term’. Some parts that today are still concepts will be prototyped. Thus more open questions on the way to a ‘consolidated building product data model’ will be answered. Others, off course, will stay open. Results from VEGA up to date are quite promising. Vega will be a mayor step towards the authors concept of ‘integration by communication’ (Junge 19991) and it is extending it to collaboration of human- inter- actors. Up till today the project was concentrated on working on the technological foundations. Future work will use these for building of a new type of applications enabling architects and engineers not only to communicate through their applications in an distributed environment modeled after their habits, it also will be a major steps to the new CAAD generation. In ‘A Dynamic Product Model’ (this Volume) indications are given about the direction these applications will take.

References

ATLAS, [1994]
ESPRIT Project 7280: various deliverables

Bjoerk, Bo-Christer [1993]
Byggproductmodeller- Nulaege
Byggforkskningraedet, R27:1993

Bjoerk, Bo-Christer [1994]
Contributions to the theory of building product data models

Bjoerk, Bo-Christer [1995]
Requirements and information structures for building product data models
VTT Publications no245, Technical Research Centre of Finland, Espoo

Eastman, C., [1978]
The representation of design problems and maintenance of their structure.
Latcombe (ed.), Application of AI and PR to CAD, Elsevier, North-Holland

Gielingh, W., [1988]
General AEC Reference Model, ISO TC 184/SC4/WG1 DOC N.3.2.2.1

IMPACT [1991]
IMPACT Reference Model. Deliverable, ESPRIT Project Impact

COMBI, [1994]
Junge, R. Ammermann, E. Katranuschkow, Scherer, R.
ESPRIT Project 6909: deliverable A2, A3, B3

Junge, R. [1991]
Integration by Communication.
1st International Symposium on Buillding System's Automation - Integration
Conference Proceedings, Univ. of Wisconsin, Madison

Junge, R. and Storer, G., [1993]
B/C Application Protocol Planning Project. ISO TC 184/SC4/WG3/T12 AEC Working paper

Junge, R, [1994]
Bauproduktmodelle: Eine Einführung
Arbeitspapier, OFD Berlin, Bundesliegenschaftsverwaltung

Junge, R. and Liebich, Th., [1994]
Produktmodellierung im Architekturbereich.
Abstracts of IKM, HAB Weimar

Luiten, B., Luijten, B., Willems, P., Kuiper, P. and Tolman, F.P., [1991]
Development and Implementation of Multilayered Project Models.
Mangin, J-C., Kohler, N. and Brau, J. (edts), Computer Building Representation for Integration, Pre-Proceedings of the second international workshop, Ecole de polytechnique federale de Lausanne, Lausanne,

Luiten, G., Froese, T., Bjoerk, B-C., Cooper, G., Junge, R. Karstila, K. and Oxman, R. 1993
IRMA, An information reference model for architecture, engineering and construction.
Mathur, K., Betts, M. and Tham, K. (eds.) Managementof Information Technology for Construction., World Scientific & Global Publication Services, Singapore 1993

Landsdown, J. [1983]
Expert Systems: Their impact on the construction industry
Report to RIBA Conference fund, London 1983

NEUTRABAS, [1991]
ESPRIT 2010 project: Outfitting Information Model, Deliverable 4.2.2

Richens, P. [1978]
The Oxsys system for the design of buildings.
3rd International conference on Computer in Engineering and Building Design CAD78., Ipc science and technology press, UK

Tarandi; V. [1993]
Oblect oriented communication with NICC, neutral intelligent CAD communication.
Mathur, K., Betts, M. and Tham, K. (eds.) Managementof Information Technology for Construction., World Scientific & Global Publication Services, Singapore 1993

Tolman, F.P. and Wix, J., [1995]
ISO TC 184/SC4/WG3/T12 AEC, Part 106 Working paper

Turner, J., [1990]
Building Systems Model. ISO TC 184/SC4/WG1 Working paper

van Nederveen, G.A. and Tolman, F.P., [1992]
Modeling multiple views on buildings. Automation in Constuction, vol 1, 1992 nr. 3

Wager, D. [1984]
Expert systems and the construction industry.
CICA, Construction Industry Computing Association, Cambridge UK