Management and Communication of Distributed Conceptual Design Knowledge in the Building and Construction Industry

Final Report

24 June 2002

dr.ir. Josef Petrus van Leeuwen

Host organisation:

Instituto Superior Técnico

Dep. Engenharia Civil

Table of Contents

1 Introduction

This document reports on the project 'Management and Communication of Distributed Conceptual Design Knowledge in the Building and Construction Industry'. This project was initiated by the author as a collaboration between Eindhoven University of Technology, The Netherlands, and the Instituto Superior Técnico, Portugal. The project was financially supported by the Fundação para a Ciência e a Tecnologia of the Portuguese Ministry of Science and Technology. This financial support has made it possible to perform this research project at the Instituto Superior Técnico and has facilitated the synergy of research experiences from the various areas of expertise of the participants involved in this project.

1.1 Research Team

The following persons were involved in the research project.

Dr. Jos van Leeuwen	TU/e	Associate professor in collaborative design at the Design Systems group; principal researcher
Prof.dr. João Bento	IST	Chair of the Information and Design Support Systems group
Prof.dr. Bauke de Vries	TU/e	Chair of the Design Systems group
Dr. José Duarte	IST	Assistant professor at the Information and Design Support Systems group
Eng. Fransisco Regateiro	IST	Computer scientist and researcher at the Information and Design Support Systems group
Dr. Sverker Fridqvist	TU/e	Post-doctoral research at the Design Systems group
Ir. Joran Jessurun	TU/e	Computer scientist and researcher at the Design Systems group

1.2 Collaborative Design

Collaborative Design is a term that designates not just the collaborative nature of design in modern construction projects; it also represents the development and application of information technology that facilitates this form of collaboration. This area roughly concerns three domains of research:

- Design Management
- Information and Knowledge Modelling
- Communication and Data Exchange

Very summarised, these areas can be described as follows.

Design Management deals with the organisational issues of collaborating in a design and construction project. It looks at various models for design planning, working in design teams, and at organisational forms of project management in design and construction.

Information and Knowledge Modelling involves finding the approach to formally store information and knowledge, in this case concerning design, that best suits the requirements of supporting design and production tasks.

Communication and Data Exchange involves the technology to transfer the modelled information and knowledge between participants in the design and construction project. Apart from having an impact on how the modelling is done, this area also develops and applies technologies such as data transfer protocols and mark-up languages.

1.3 Focus of this project

The focus in this research project is initially on aspects from the latter two domains mentioned above: Information and Knowledge Modelling and Communication and Data Exchange. The intention is to improve the state of the art in these areas such that Design Management, the first mentioned domain, can be better facilitated.

Information and Knowledge Modelling for design support has to deal with design as a problem solving process, in which *both* the problem and its solution are to be discovered in an unpredictable manner. The ways of dealing with information during the design process cannot be prescribed without intrusion on the freedom of creativity experienced by designers. Computer support systems for a relatively chaotic process like design must allow the users of such systems to have sufficient influence on their way of working with the system. This means that the designer must have a high level of freedom in the way information describing the design is defined.

The Communication and Data Exchange part of the problem addressed in this research involves the exchange of design information and design knowledge both within the scope of a design team or between individual designers who aim to share this knowledge. Sharing data in a multi-user asynchronous environment means that externally accessible repositories must be used for storage of data, which can still be either centralised or distributed.

The information modelling approach that is used in this project is derived from the PhD thesis of the author (van Leeuwen, 1999). The approach is called the Feature Based Modelling (FBM) framework. In practical terms, this project addresses the problem of sharing design information and design knowledge that is modelled using the approach of the FBM framework, through utilisation of Internet techniques and Case-Based Reasoning.

2 Research Background

2.1 Sharing Design Knowledge for Collaborative Design

Collaboration in design is one of the key factors of successful design in building and construction. It is necessary in virtually all stages of design and involves a great variety of disciplines and partners in the project. This includes the multiple design partners who each bring in their own domain-specific knowledge, the client and users of the future building, contractors and suppliers in the building and construction industry, and prescribing and regulating authorities.

Probably the most important aspect of collaboration is sharing knowledge. Design knowledge is the totality of on the one hand information about a design, including the data that describes the design and the contextual meaning of this data, and on the other hand information about how a design is achieved and evaluated. We can distinguish design knowledge into the kind of knowledge that is particular to a design project and the kind of knowledge that is used in projects but is of a generic nature.

A third kind of design knowledge is design cases. Knowledge from previous designs is continuously built up, consciously or unconsciously, in the minds of designers and applied, again consciously or unconsciously, in new design projects.

Digital media have greatly improved the efficacy of collaborative design, by reducing the effort and faultiness of communicating design information. Despite that, prevailing ways of exchanging digital design information are often semantically poor or semantically incorrect and lead to mistakes in interpretation, by humans and by computer-systems. However, digital media promise more efficient means to express and communicate design knowledge. An increased semantic level of design data may help to reduce faults and therefore will increase the efficiency of the collaborative design process. An example of how this can be achieved without too many compromises to the requirements posed by the nature of design, regarding creativity and the many different ways of addressing design problems, is the work by Hendricx & Neuckermans (2001), Hendricx (2000), and Geebelen & Neuckermans (2000).

2.2 Formalised Design Knowledge

One way to increase the semantic level of design data is to develop semantically more detailed and more explicit standards for data exchange. The Industry Foundation Classes (URL 1), under development by the International Alliance for Interoperability, have a great potential to become the de facto standard for data exchange exactly because they will bring the data exchange to a higher semantic level.

Innovative designs, however, incite the need for expressing novel concepts that cannot be described using the standards. In these cases, designers may feel limited by standards since they only provide means to express design intentions on a generic level that cannot catch much of the specific intentions. To fully support design, modelling tools need to allow designers to define design concepts that exactly represent the rationale of the design (van Leeuwen, 1999, de Vries et al., 2001, van Leeuwen & Jessurun, 2001). This would make such a design support system specially tailored for a particular designer.

To formally define design concepts is a form of knowledge modelling, since the designer expresses not only the actual design case, but also the concepts used in the design. The concepts represent the body of design knowledge that was used to come to the particular design solution, and they can be reused for other designs. The issues of developing design systems that are sufficiently dynamic and flexible to support design tasks have also been addressed by Galle (1994), Ramscar (1994), Junge (1995), and are subject of research in the work by Eastman et al. (1993, 1995, 1997), Ekholm & Fridqvist (1998, 2000), and Fridqvist (2000).

Formalised building product information is another way to enhance the semantic levels of design data. Although product information is increasingly often made available digitally, the format is mostly ill structured, such as a web page with text and images that can only be interpreted by human readers. To search this kind of format is very time consuming but can be greatly enhanced if the information structure allows automated searches (Bakis and Sun, 2000). Once product information is made available in a structured way that allows computers to interpret the content, it can form a much more valuable source for design support systems in providing intelligent feedback and suggestions to the designer (Augenbroe, 1998, Jain & Augenbroe, 2000). Again, standardised models will play an important, but limited role in the formalisation of product related design knowledge. The role is limited for two reasons: firstly, standards, in the way they are currently developed, cannot be expected to both reach sufficient level of detail and to remain sufficiently generic for the required general applicability that they are developed for. Secondly, new products, materials, and construction methods will continuously appear, which will require additions to any but the most generic standards.

3 Research Objectives

This section repeats the research objectives as described in the original proposal.

The main objective of the research project is to develop a technological and organisational framework for the management and communication, in the building and construction discipline, of conceptual design knowledge in a distributed environment. This framework allows designers to model their expertise and use it in computational design reasoning; it also allows them to share this expertise, enabling them to use each other's design knowledge and reasoning mechanisms. The project aims to develop the technology and the organisational aspects that are required for designers to build up a common, but distributed, design knowledge-base.

Management and Communication of Design Knowledge

Design knowledge is knowledge about, e.g., design problems, design concepts, design decisions, design methods, design solutions. This may be knowledge about a particular design case or generic design knowledge describing the typology of a set of cases. Management of design knowledge involves a number of aspects: acquisition, formalisation, and modelling of knowledge, making it available for digital processing; storage and retrieval of knowledge, this also includes search and comparison mechanisms; and the reasoning mechanisms themselves, e.g., case-based reasoning or rule-based reasoning.

Communication of design knowledge allows the exchange of knowledge, e.g., between participants in a design project, and the distribution of knowledge, e.g., across a network of a design discipline. In either case, the design knowledge is made available for others to use inside or outside the initial context of the knowledge.

Technological Aspects

The technological aspects of the framework concern:

- Integration of an information modelling technique with the information processing mechanisms available from the area of artificial intelligence. The modelling technique will be based on the approach developed in van Leeuwen (1999), which was designed to deal with user-defined information structures.
- Management of design knowledge. This includes the aspects of storage and retrieval of both declarative and procedural knowledge.
- Usage of design knowledge in reasoning mechanisms. Implementation of AI technologies, with a focus on case-based reasoning.
- Communication technology. The development of the elements of the distributed environment, based on Internet technology.

The technological aspects of the framework will be based as much as possible on existing technology, but are likely to require development of additional software.

Organisational Aspects

Application of the above described technology in a practical environment introduces the following organisational aspects:

- Protection of ownership of disseminated design knowledge. Since the knowledge is to be made available through Internet, the protection of ownership is likely to become a problem and will ask attention on the organisational side in addition to technological security aspects.
- Responsibility for knowledge. Usage of the common knowledge-base will raise the question who is responsible for the validity of the reasoning mechanisms and design knowledge 'borrowed from the network'.
- Implementation of the environment in practice. The research project will be developed in a scientific setting, however, the implications for implementing it in design practice will be part of the study.
- Testing and evaluation. Within the scientific setting, the developed framework will be subject to testing and evaluation. Here, the project will be narrowed to the sub domain of structural design. This will involve international collaboration with experts in a variety of areas besides structural design, including artificial intelligence in design and product modelling in the building and construction industry.
- Study of the feasibility of practical application. This study will investigate whether the benefits of using this technology in various settings (project scope, discipline-wide, inter-

disciplinary) will countervail against the required efforts in terms of learning curve and extra resources.

4 Methodology

This section describes the methodology followed in this project to achieve its goals. It briefly introduces the information modelling approach that is used as the basis for the developments and then describes what technologies have been investigated and integrated with this approach in order to arrive at a feasible system design and its implementation.

4.1 Dynamic Feature-Based Modelling

The system for sharing design knowledge that is described in this report, is based on a dynamic approach to product modelling that is inspired by *Feature-Based Modelling* (FBM) in mechanical and structural engineering. The theory for this approach in the building and construction context, called the *FBM framework*, has been described in (van Leeuwen, 1999) and can be characterised as a property-oriented modelling approach (van Leeuwen et al., 2001). Following are the main issues addressed by the FBM framework:

Property oriented

Rather than defining object-classes to represent building components that embed many properties, this approach takes the properties themselves as a starting point for modelling and allows the designer to compose the object-representations from properties. In fact, the FBM framework has no need to distinguish objects from properties, calling them both features. It allows a feature to be used with different roles: as a key modelling object in one situation and as a property of another feature in another situation. For example, a feature 'Spatial function' can at one stage in the design process be a key object and at a later stage be changed into a property of the 'Space' feature where this function is performed.

Flexibility in modelling

Properties have an independent existence, independent of the object that carries the property, and they can be shared by multiple objects. The existence of a property such as 'load-bearing' is not dependent of the building component that performs this task. This not only allows the designer to describe the design rationale in a natural way, it also better supports the process of design, because it can deal with the often inconsistent and 'composing' nature of design. The load-bearing building component might be removed from the model while the property 'load-bearing' is still required and needs to be transferred to another component that might be added to the model at a later stage.

Ad-hoc modelling

Generic information about design concepts is described by feature types. They are the typologies that a designer uses in reasoning about the design. Specific information about a particular design case is modelled in feature instances that are created on the basis of the feature types.

The FBM framework does not constrict the modelling of properties of feature instances and relationships between them to those that are defined in feature types. Feature instances in the model can be changed as needed, without prior adjustment of feature types. For example, if the context of a building design requires a 'door' to have the property 'fire-resistance', but this property is not defined for the door-type, then the designer still can add the property to the model, without having to modify the door-type. This approach reflects the actual situation that is often encountered in design, where specific features are added to typical design solutions, or where components are used in unforeseen ways.

Designer-defined typologies

In addition to this flexibility of building relationships between objects and between objects and properties, designers can also define new typologies, i.e. new feature types. This capability allows designers to generalise design concepts that result from a particular design case, or to formally describe concepts that represent general aspects of their design methodology. These custom-defined feature types build up a library of concepts that represent the designer's specific knowledge. The extendibility of the conceptual model also serves the need to define new typologies, e.g., for new construction products or construction methods.

The FBM framework includes simple data types and complex data structures, but also constraints and procedural types. Although the latter two have not been implemented fully yet, they do form an integral part of the framework. The framework implements the property-oriented system and its flexibility and extendibility by defining an object model with a set of meta-classes for both feature types and feature instances. The system's end-user appears to be dealing with feature instances as objects that are instantiated from the feature types, but in fact, both feature instances and feature types are objects that are instantiated from the meta-classes. This way, the system provides run-time extendibility of the schema, while keeping all objects consistent with the meta-classes and maintaining the relationships between instances and types. In addition, the ad-hoc modelling capability, which allows deviations of feature instances from their types, is provided through the meta-classes.

4.2 Case-Based Reasoning

Case-Based Reasoning (CBR) plays a dual role in the context of this project. In its first role, CBR supports the designer in local activities by means of a technology that is called *Feature Type Recognition* (FTR). Structures of feature types and structures of feature instances are compared and, using a set of heuristics, conclusions are drawn on the similarity of these structures. When a network of instances in the model matches the structure of a known type, this can lead to the conclusion that this network can be replaced by the structured instantiation of the type. As a result, the model can be enhanced by the additional information that is related to the recognised type, giving a richer meaning to the model.

For example, a number of interrelated 'wall' instances in the model can be recognised to form an enclosed space. This structure of walls can be matched to a 'space' typology that defines walls as its enclosure. If the designer accepts the suggestion that the walls can be recognised as an enclosed space, the system can enhance the model with additional information about spaces, such as volume, area, function, temperature, etc.

The heuristics and reasoning mechanisms used for the comparison can be more or less restrictive. A more restrictive search would result in more exact matches between the network of instances and the structure of the found types. Less restrictive searches would allow less exact matches to appear as well; this makes it possible to find similarities based on the typologies of components. For example, it would allow a 'space' instance that has relations to a number of 'desk' instances to be recognised as an instance of the type 'office', even if the relations between the 'space' and the 'desk' instances in the model do not appear exactly in the same way as they are defined by the type 'office'. In other words, the system can be tuned to treat similarity of properties and relationships in a more or less fuzzy manner.

A second key role of CBR lies in the targeted functionality of *sharing* design knowledge. Here the recognition mechanism as described above is applied in a similar fashion but design knowledge is searched across the Internet as well. This technology can be applied by designers in the context of particular design projects, but also outside project scope. The matching algorithms provided by FTR allow designers to search the Internet for applicable design cases and for formalised descriptions of design knowledge.

The study of CBR for application in this project concentrates on two aspects. Firstly, the matching problem itself is subject of investigation. This part of the study deals with the various approaches to compare relationships between features on the basis of the characteristics of these relationships. Relationships between features are characterised by information about the respective features and by information about the role that they play in the relationship. Both these kinds of information are taken into account in the process of matching structures of feature data.

Secondly, the interpretation of the outcome of the matching algorithms is subject of investigation. Since the reasoning mechanism can subject a large number of feature types and a variable selection of feature data in the model to the matching process, the result will consist of a potentially large number of 'candidate' types that must be sorted according to their level of matching. The interpretation of what is the level of matching and the mechanism for sorting the candidates is not trivial. Not only does it depend on the quantitative results of the matches, but also on the meaning that the designer ascribes to the matched (and missing) properties and relationships.

4.3 Collaboration Facilitated by Internet Technology

Internet technology is increasingly often used to facilitate collaboration in design projects. Several forms of Internet technology are applied for this purpose. Currently the most popular technology is provided by so-called Electronic Document Management systems (EDM). These systems provide a central repository for the storage of documents related to a team of users that have access to this repository. The repository is central in a network that is either a LAN network, an Intranet, or an Extranet. Many systems provide an web-based user interface and a dedicated client application with enhanced facilities. Users of the system can upload and download documents and open each other's documents for reading, editing, and redlining. The system often provides versioning functionality to keep track of changes and different versions of documents.

Although these systems are becoming current practice, there are a number of issues that make these systems not as attractive to practice as they might seem at first glance. The three most important of these issues are: the fact the all documents are stored centrally; the limitations of documents as storage media; and the limitations of document-based versioning facilities.

Centralised document repositories (see Figure 1) are problematic in practice, since they do not reflect the structure of collaborating organisations that are using these repositories. Collaborating teams are normally formed by co-workers from many different corporations that have both shared and individual interests and responsibilities. Sharing information by storing it in a central repository is not what these professionals would prefer, since it forces them to 'give away' the ownership of documents and the responsibility for safeguarding them. Even if, technically and legally, issues of copyright and ownership could be dealt with, strategically an organisation would not likely be happy to centralise all corporate knowledge together with other organisations. An additional issue is that teams of collaboration, especially in the construction industry, are short-lived and manifold. This means that each collaborated project would require its own central repositories, which leads to a huge amount of redundant storage of corporate knowledge.

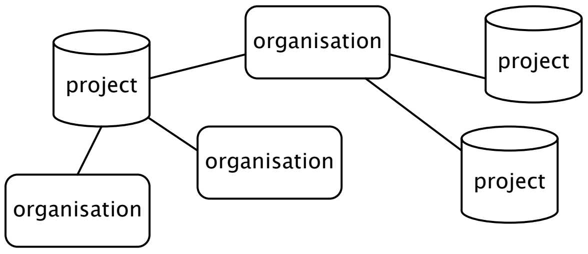

Figure 1. Project-based central repositories.

This project has developed an alternative approach to sharing information. This approach does not rely on central repositories for a project and does not use documents as the basis for storing information. The point of departure in the proposed approach is a peer-to-peer network of nodes that provide distributed repositories of information and that allow remote access to the information to partners in the collaboration process. The configuration of the network of nodes is not fixed but can be tailored to the requirements of the team and its members. This means that the decisions about where information is stored, who keeps the ownership of it, who is responsible for the storage, validity, and access to the information are again part of the agreement of collaboration and not implied by the tools used for the collaboration. Conceptually, this puts the collaborating organisation in the centre of the technology, not the project. In terms of management, both information management and organisation management, this is a much more logical approach. Figure 2 shows a configuration where each organisation maintains its own repository and where collaboration on projects is facilitated by providing partners in the projects with access to certain parts of the corporate knowledge. The flexible nature of the system of networked nodes makes other scenarios possible that are relevant in collaborative work. Nodes in the network can be set up to represent any logical domain of information, be it related to an individual organisation, a branch-organisation, a governmental institution, a specific project, a group of projects, long-term or short-term collaborations, etc.

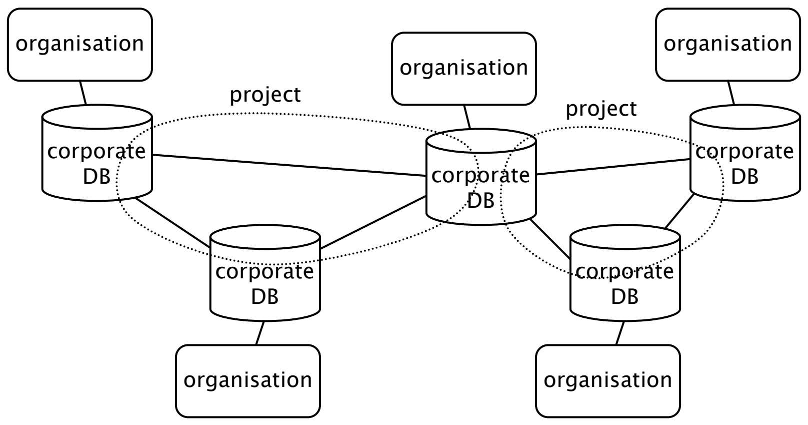

Figure 2. Corporation-based distributed repositories.

Information is organised throughout the network on the basis of objects that reside in namespaces, not in documents. This is a change of concept for the practice of many disciplines, but it offers great flexibility and helps to organise information in a logical manner. As a result, project information consists of interrelated objects that are distributed over the various nodes in the network where they reside under the responsibility and ownership of their creators. The assortment of objects in namespaces provides a mechanism to organise data in a logical manner that at the same time ensures identity and a way to unambiguously locate data. Contracts and agreements will be based on the state of objects in namespaces rather than documents.

Strongly related with the notion of organising data in objects within namespaces is the notion of versioning. Versioning is maintained on a per object basis and the version information of an object becomes part of the object's identity. Objects in this context are feature types as well as feature instances, meaning that generic design concepts will be organised in versions just like the actual design data in feature instances is organised in versions. Version control of objects is an essential part of keeping data in a distributed modelling environment consistent and becomes even more important when users are given access to typological specifications of data, as is done in the FBM approach.

5 System Specifications and Design

The functionality of the Feature-Based Modelling approach is used to build so-called *Design Knowledge Servers* (DesKs). The DesKs manifest an Internet-based client-server technology that is to be applied in a number of scenarios of sharing distributed design knowledge.

5.1 Targeted Applications and Usage Scenarios

Design Knowledge Servers function as a network of interrelated servers that provide managed access to distributed data. Such a network can be used inside and outside the scope of a design project. Inside a project-scope, each node in the network represents a repository of design knowledge that falls under the responsibility of the owner of the node, e.g. a structural engineer or an architect.

Outside the scope of any particular design project, the DesKs can be used to provide access to 'general' design knowledge and public design information, such as product data, information about construction services, design methods, public evaluation tools, etc.

Below, four example scenarios are described, in which the DesKs can be used to share design knowledge.

Collaborative Design Projects

The Design Knowledge Servers can be used in this scenario to develop design solutions by gradually building up formal definitions for design concepts (specific feature types) in combination with the application of more standard design concepts (generic feature types). The project's design database, containing both feature types and feature instances, does not need to be centralised but can be distributed over the network of systems managed by the collaborating partners. This way, the participants maintain their ownership and responsibility of the knowledge and data involved in their design task. Other participants are granted access to design data based on their role in the design team, on contracts, and on shared design tasks and responsibilities. Knowledge that is external to the design project, in the form of modelling standards or automated code-checking procedures, can be included in the DesKs network if the communication and data exchange protocols are supported.

Design Knowledge Commerce

In this scenario, there is a contract between a party needing a design solution to a particular problem and a party able to provide the solution. The parties have been brought together through a search session provided by one or multiple DesKs, where the searching party has input the design problem into a search algorithm and the providing party had made the design solution, or information about how to achieve a solution, available on a server in the network. Based on the contract between the parties, rights are assigned to the acquiring party to use the provided design knowledge.

This commercial scenario involves many complications in terms of copyrights, multiple uses of the provided knowledge, re-use of knowledge provided by third parties, composite solutions, etc. These complications need to be addressed before this scenario can actually function in a commercial setting. However, a similar scenario can be envisioned in the context of, for example, designers' associations. Designers subscribe to the provided services, acquiring the right to use all knowledge available while feeding the association's servers with their own knowledge.

(Historical) Design Reviews

The ability for design critics and architectural historians to make their observations of design rationales in historic or new buildings available in a formalised manner would allow them to share this knowledge with the architectural design discipline. This would offer a tool for very direct feedback into today's design practice. It would also support comparisons of design approaches and alternative design solutions.

Product Databank

Providing product data in a ready-to-use format is a very feasible application area for computer support in design. The construction industry has already seen several successful developments in this area. For example, the Dutch building-documentation system (NBD) provides this kind of information in a digital format, although this format often is too loosely defined to be utilised directly in computer supported design tasks. The international developments on standardisation of Part Libraries (ISO 13584) aims to provide a more detailed definition of how part information is to be provided. The part Libraries deal with structural, descriptive, and procedural knowledge in electronic catalogues to serve the selection, evaluation, and representation of parts in engineering tasks (Pierra, 1997).

The scenario of applying DesKs for providing product data to the building design community follows a similar path: manufacturers and suppliers can provide their own specific typologies and samples of products and materials in feature types and instances. Product information made available through Design Knowledge Servers can be used directly in the design process.

5.2 System Design

Targeting the application scenarios above, a system has been designed that provides the FBM framework functionality through an object-model that supports remote access to distributed data sources. The FBM framework implementation is used in the development of two applications:

1. The DesKs *WebServer* application allows publication of design knowledge through a common web server and makes the design knowledge available in two ways: as HTML web pages and as a Web Service. HTML access to the knowledge is limited and provides only a browsing type of digestion of the provided information. The Web Service access is more enhanced and allows client applications to consume search, storage, and retrieval services.
2. The DesKs *WebNode* application is a modelling tool that provides all the FBM capabilities of formalising concepts and dynamically modelling designs. The application also provides dedicated client functionality to access the Web Services provided by the DesKs WebServer application. This enables the application to search and retrieve remote design knowledge and to actively work with distributed design data.
   In addition, the WebNode application provides peer-to-peer functionality that enables it to share its data with other WebNode instances in a wide area network. This part of the application is not based on common web server technology, but does use the common protocols HTTP and SOAP.

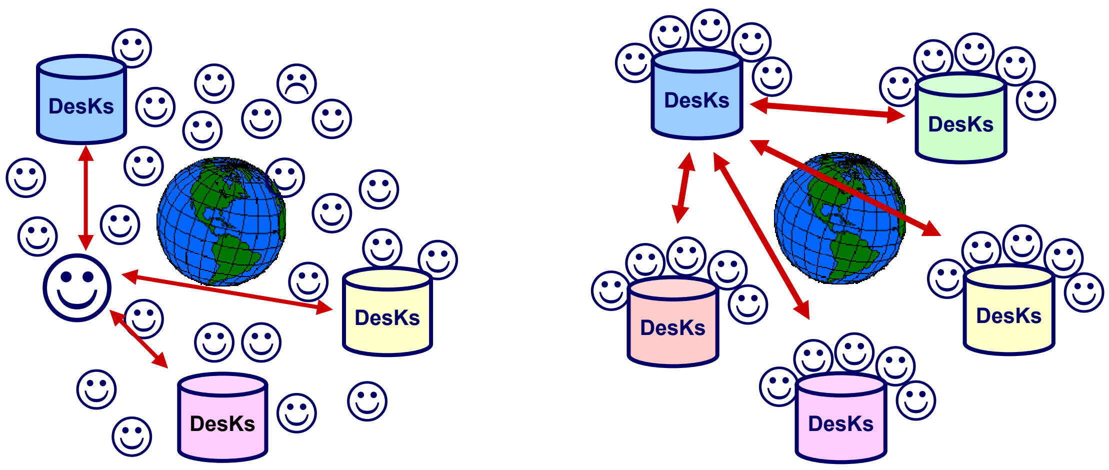

Figure 3. On the left: DesKs WebServer application for browser or dedicated client access.
On the right: DesKs WebNode application for peer-to-peer networking.

Object-model and meta-layer

The FBM framework is implemented as an object-model that provides a meta-layer of classes defining the kinds of feature types and the kinds of feature instances that can be modelled. The run-time system thus contains three layers of data: the bottom layer that defines the actual design data in terms of feature instances, the middle layer that defines the typologies used for creating the instances, and the meta-layer at the top that defines the 'syntax' for the lower two layers.

The meta-layer includes classes for simple data, such as strings and floating point numbers, and complex data structures. Complex data structures are defined as features that contain components. At the type-level, a component of a complex feature type is a reference to another feature type with information about the role and occurrence of that type in the context of the complex type. At the instance-level, a component references one or several other instances. The instance-level component can either be instantiated from a component defined by the type of the instance, or it can be a 'custom' component that is added ad-hoc to the feature instance.

Components are specified by a role-name and a role-type. The role-type distinguishes associations between features, decompositions, and specifications; the latter being a particular kind of association that tells us that one feature specifies a characteristic of another. These role-types add semantic meaning to the model that is used in the reasoning support procedures that are under development for the Feature Type Recognition functionality.

Version management

An important issue in collaborative activities is how to control versions of information. Keeping track of versions of information serves three objectives: to record the history of information in order to allow undo-operations; to allow changes to data without compromising references to previous versions of that data; and to make it possible to inspect and compare versions.

Current practice document management systems provide version control, but only at the document level. For collaborative design, version control is required at a finer level of detail for a combination of reasons. The number of people working with design data is large, the total collection of design data is large, documents are not always the basis for storage, and perhaps most importantly, there are strong relationships between chunks of data, within documents or crossing the scope of documents.

The FBM framework has strong support for version control of both feature types and features instances. Editing of feature data (both types and instances) takes place via a checkout-and-commit mechanism, through which users get temporary editing privileges. While data is checked out for editing, previous versions can continue to be used. After editing, data can be either submitted as a new version, or committed as a revision. Revisions of feature data are inferior to versions in the sense that they cannot yet be actively used in modelling operations, only for further editing of the data. This reduces the number of versions and allows distinction of which submissions are of real interest and which have only an intermediate status. Only revisions of the latest version are backed up by the system.

Versions are distinguished by the combination of a major version number M and a minor version number n in the form M.n. New version numbers are incrementally assigned upon submission and minor version numbers are reset to zero after the submission of a new major version. Submitting a version to the system can lead to a new major version or a new minor version. Minor versions indicate backwards compatibility, which means that the version can also be used in place of previous minor versions of the same major version. For example, adding a property to a feature type leads to a new minor version because it does not compromise the functionality of the type in places where the type without that property was expected. New major versions are not backwards compatible, meaning that they cannot be used in place of any preceding versions. Modifications such as removing properties or changing the type of properties will generally lead to new major versions. Whether a submission is a new major or minor version, is determined in the first place by the user. However, the system will enforce major versions when it detects backwards incompatibility. Upgrading in instance to a more recent minor version of its type is generally possible and can probably be done automatically, although this functionality has not yet been studied in detail. An incremented revision number is assigned after each time a revision is committed or a version is submitted; the revision number uniquely identifies a revision or version of the feature data.

Identification and namespaces

In the original definition of the FBM framework (van Leeuwen, 1999), feature types were located in feature type libraries, while feature instances that represented a design case were collected in feature models. Libraries and models were defined as single storage locations (e.g. a single database or single data file). In the current implementation of the FBM framework, libraries and models are replaced by namespaces, in accordance with the usage of namespaces in XML. Namespaces have the advantage that data can be distributed over multiple resources (although a mechanism for retrieval of all parts must be available) and namespaces can be related to a URI reference (Uniform Resource Identifier), such as a URL, giving globally unique identification to all names that are unique within the namespace.

The notion of namespaces is applied in the current FBM framework to identify collections of features, both types and instances, that together form a particular body of design knowledge, or that represent a particular design case.

Ownership, authentication and authorisation

Each individual feature type or feature instance is owned by an identifiable user. Users are identified by their email address and are authenticated using a password. Each initial access to an application of the FBM framework will require authentication. Users have full access rights to the features they own and can grant anonymous access or access rights restricted to other users or groups of users. Authorisation will take place automatically upon each access. Namespaces have owners as well and can have restricted access. Access rights set for individual features in a namespace impose restrictions further to those that are set for the namespace as a whole.

Groups of users can be defined to represent teams in collaboration projects or to specify other kinds of group access to certain data. User can acquire access rights through membership of a group, but higher individual rights will not be restricted by such membership.

While the authorisation mechanism is still under development, the following levels of access rights are currently distinguished, listed in incremental order:

- Copy (read but only for copy, not for reference)
- Read (read but not instantiate)
- Instantiate (relevant for types only)
- Modify (change contents but not add)
- Add (add contents)
- Write (includes delete and rename)
- Ownership (includes the right to set access rights and to transfer ownership)

Access in a distributed environment

A previous implementation of the FBM framework supported access to remote data by offering the capability to download feature data from URL's. This approach only supported read-access to the remote data and thus solved only a small aspect of the collaboration problem. It did not support real-time collaboration in any way.

The current implementation of the framework supports direct remote access to data. Together with the mechanisms for authorisation and checking out data, this provides the means to collaborate in a distributed environment; distributed not only in terms of distributed users, but also in terms of distributed data. Users can access remote data as if it were local data, albeit that they are subject to the authorisation settings of the remote system.

Having the option to distribute data, project managers can now decide to leave the physical ownership of data where it belongs: with the experts that are responsible for it.

6 Implementation Issues

The implementation of the FBM framework and the Design Knowledge Servers is made using Microsoft's .NET framework and the C# programming language. The functionality of the FBM framework is implemented into a core module to which the DesKs applications are connected.

The framework's object-model forms the programming interface to the core module for the development of applications. Internally, the object-model is persisted into a relational database (current testing uses MS-SQL server). For communication with other applications, an important feature of the core module is the import/export capability from and to XML documents. Feature types can be streamed from and to XML-Schema's, and feature instances can be streamed from and to XML documents. Both schema's and documents are validated by a generic XML-Schema that represents the syntax of the FBM framework. The XML documents containing the feature instances are also validated by the XML-Schema's that contain the respective feature types.

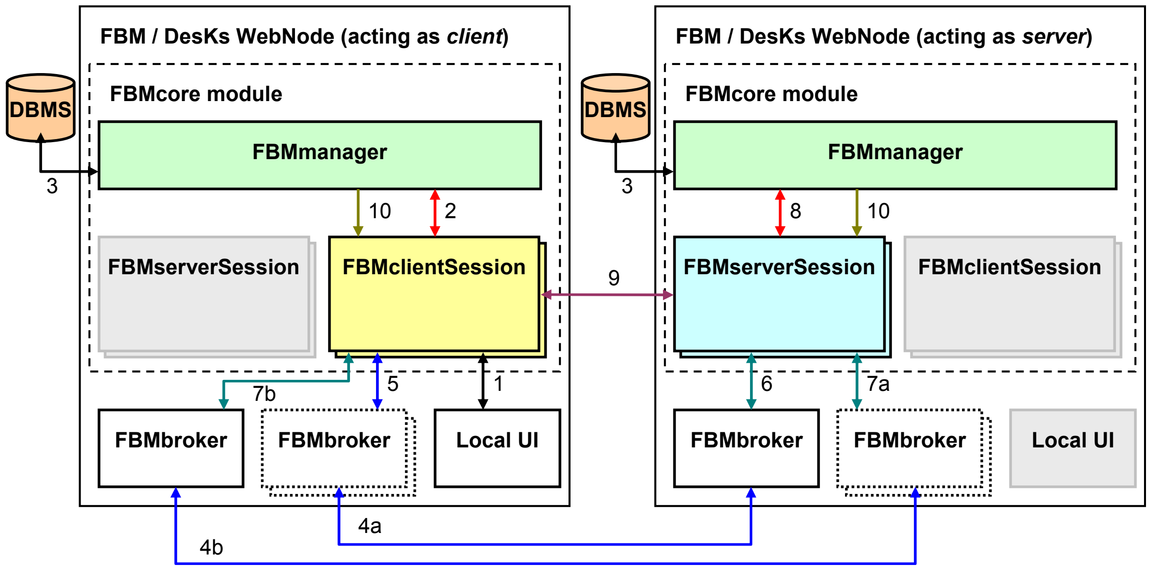

Figure 4. General architecture of the DesKs WebNode application.
On the left an instance that is acting as client, on the right one that is acting as server.

Networking DesKs

The peer-to-peer functionality of the DesKs WebNode application is built using the .NET framework's *remoting* facilities. Simply put, remoting allows objects on a server to be accessed by remote clients, as if they resided in the local memory of the client. A WebNode client can have open connections with multiple servers, which allows the simultaneous utilisation and combination of feature data from various resources. Vice versa, servers allow multi-user access and provide functionality for sharing sessions on a server. Sharing sessions allows teams to work together with the same set of namespaces and features retrieved from servers. A subscription mechanism notifies client applications about changes at the server, to avoid problems with outdated information at the remote clients.

Figure 4 shows a general picture of the architecture of the DesKs system as it is currently implemented in the prototype WebNode application. The figure also indicates, in a simplified manner, the communication lines between the various parts of the system, as listed below.

1. The FBMcore module is prepared for multi-user access, either local or remote. Each local user of the application communicates with a private client-session object.
2. Client-session objects communicate with the single manager object in the application, which provides the object-model of the FBM framework, including objects for namespaces, feature types, and feature instances.
3. The manager instantly persists all modifications through an OLE-DB interface.

4. The initial contact (4a) with remote servers is made through a broker object that exists at the remote server and for which a proxy is created at the client-side (proxies have dotted outlines in the figure). Each instance of the application runs a single broker object, but can maintain proxies for multiple server-brokers. This connection is two-way: the server establishes a connection back to the client (4b) using a proxy for the client's broker object. This second connection is used for communications initiated by the server[1] (see also item 7 below).

5/6. Once the connection with the server is established, the client-session can retrieve information about the server via the broker-proxy (5). For each remote client, the broker at the server creates a server-side session (6) with which the client can communicate as if the server-side session were a client-session to locally managed data (see also item 9).

7. When a client establishes a connection to a server, the server creates a server-side proxy for the broker of the client in order to push data back to the client on its own initiative. Via the server-side proxy (7a), the server can get access to the client-session (7b). This way the server can notify the client to retrieve updates for feature data to which the remote user has subscribed.

8. The server-sessions communicate with the server-side manager in the same way as client-sessions communicate with the local manager (see item 2).

9. After the brokers have been used to establish the connections between client and server, and once the client has connected to its server-side session and vice versa, proxies will exist on both sides for the remote session objects. The broker is no longer needed for communication that is initiated by the client. Remote feature retrieval and editing activities related to remote features will now be executed directly between client-session and server-session, using proxies for remote sessions and for feature data that resides in the server-session.

10. Users need to be notified of modifications of data made by other users. This is done through a subscription mechanism that registers users' interests in certain data on a per session basis. Upon notification by the manager, client-sessions inform the local UI and server-sessions inform remote client-sessions of the modification events.

7 Conclusions

7.1 Results after the First Year

The immediate result of the project in its current state is a prototype for the DesKs WebNode application that is now partly operational.

In the first year of this project two major studies have been undertaken. Case-Based Reasoning technologies have been studied; in particular the techniques of pattern matching and graph theories have been investigated and further developed for the specific context formed by the information modelling techniques in this project. These development have not been finalised and are only beginning to be integrated in the aforementioned prototype.

The second study concerned the investigation of alternative approaches for remote access to data and processes using the latest technologies. Three major developments in this area have been studied (COM / CORBA, WebServices, and .Net Remoting) and one has been selected for application in the prototype development (.Net Remoting). This part of the prototype development has been carried out to a fair level of detail and operation. The FBM framework technology has been ported to the .Net platform and all aspects of distributed data-access, multi-

[1] The reverse communication from server to client could also have been implemented using remote event handling. However, experiments have demonstrated that remote event handling does not function properly in all WAN configurations.

user access, object-based versioning, and persistence are currently reaching their final stages of implementation. XML data exchange has been fully investigated and initial prototypes have been developed and tested.

7.2 Outcome and Future Development of the Project

The project was planned for a time-span of 2 years and included two cycles of system development and testing. The first year of the project has now finished and has developed very successfully and on schedule. A detailed study of the technologies required to achieve the stated objectives has been carried out; the requirements, functional design, and technical specifications of the system's architecture have been described; and the implementation of the system is well under development. The project has, however, been discontinued in its current form. The only reason for this is that the principal researcher and author of this document has accepted a new function at Eindhoven University of Technology that does not allow fulltime continuation of the research project. The project will be continued and rescheduled as part of the author's ongoing research activities.

The initial and main objectives of this research project have been achieved. Collaboration between the research groups at both institutions involved in the project has been established and has delivered good results. There has been a fruitful exchange of experiences, expertise, and knowledge between the participants in the project. The project has delivered its targets as planned and a sound basis has been laid out for the further development of the theory and of its application in practice.

The collaboration between the research groups has also been extended outside the scope of the current project. In a joint effort of two members of the research team, a special session has been organised on the research topic of this project at the renowned international conference on *Concurrent Engineering*, which will be held in Cranfield, UK, end of July 2002. The session is titled *Design Knowledge Sharing through Internet Application* and will host 6 presentations by researchers from 3 continents.

Through this research project, the basis has been laid for future collaboration between the two groups in research as well as education in Design Support Systems. Opportunities to realise this collaboration will be explored on multiple levels through direct co-operation in ongoing educational and research activities, by way of student and teacher exchanges, and possibly through joint participation in large research projects such as the EC funded FP6 projects.

8 References

Augenbroe, G. (1998) *Building Product Information Technology*. Executive white paper, Construction Research Center, Georgia Institute of Technology, 1998.

Bakis, N. and M. Sun (2000). "Intelligent broker for collaborative search and retrieval of construction information on the WWW." In: Gudnason, G. (ed.), *Proceedings of CIT2000, International Conference on Construction Information Technology*, June 28-30, 2000, Reykjavik, Iceland.

de Vries, B., et al., (2001). "The VR-DIS Research Programme, Design Systems group". In: *Proceedings of the Computer Aided Architectural Design Futures Conference 2001*, 8-11 July 2001, Eindhoven University of Technology, The Netherlands.

Eastman, C. M., Chase, S. C., Assal, H. H. (1993). "System architecture for computer integration of design and construction knowledge." *Automation in Construction* 2 (1993), pp 95-107

Eastman, C.M., Jeng, T.S., Assal, H.H., Cho, M.S., and Chase, S.C. 1995, *EDM-2 Reference Manual* (University of California in Los Angeles, Los Angeles)

Eastman, C.M., Parker, D.S., and Jeng, T.S., 1997, "Managing the integrity of design data generated by multiple applications, The theory and practice of patching", *Research in engineering design* 9: 125-145

Ekholm A. and Fridqvist, S. (1998). "A dynamic information system for design applied to the construction context". In: *Proceedings of the CIB W78 workshop The life-cycle of Construction IT*. June 3-5, 1998 in Stockholm, Sweden.

Ekholm, A. and Fridqvist, S. (2000). "A concept of space for building classification, product modelling, and design." *Automation in Construction* 3 (2000) pp. 315-328.

Fridqvist, S. (2000). *Property-Oriented Information Systems for Design, prototypes for the BAS·CAAD System*, PhD thesis, Lund University, Sweden.

Galle, P. (1994). "Specifying objects as functions of attributes: Towards a data model for design". In: Lasker G E (Ed.), *Advances in Database and Expert Systems*, pp 69-73. Windsor, Canada: The International Institute for Advanced Studies in Systems Research and Cybernetics.

Geebelen, B., Neuckermans, H. (2000). "IDEA-l, an Early-Stage Architectural Design Tool for Natural Lighting." In: *International Building Physics Conference, Tools for design and engineering of buildings*, Eindhoven, Sep. 18-21, 2000, pp.275-282.

Hendricx, A. (2000). *A Core Object Model for Architectural Design*, PhD thesis, K.U.Leuven University, Belgium.

Hendricx, A., Neuckermans, H. (2001). "A model driven approach to the development of an architectural object model." Accepted for publication in the *International Journal of Artificial Intelligence in Engineering*, Special issue on Product Modelling.

Jain, S. and G. Augenbroe (2000). "The Role of Electronic Product Data Catalogues in Design Management." In: *Proceedings of the CIB W96 Conference on Design Management in the Architectural and Engineering Office*, May 19-20, 2000, Atlanta, Georgia, USA.

Junge, R. (1995). "Aspects of new CAAD environments." *CIB proceedings, publication 180, CIB workshop on computers and information in construction*, Stanford 1995.

Pierra, G. (1997). "Intelligent electronic component catalogues for engineering and manufacturing." In: *Proceedings of the International Symposium on Global Engineering Networking GEN'97*, April 23-24, 1997, Antwerp, Belgium.

Ramscar, M. (1994). "Static models and dynamic designs - an empirical impasse vs. an inductive solution." *Proceedings of 1st European conference on product and process modelling 1994*, Dresden.

van Leeuwen, J.P. (1999). *Modelling Architectural Design Information by Features, an approach to dynamic product modelling for application in architectural design*, PhD thesis, Eindhoven University of Technology, The Netherlands.

van Leeuwen, J.P. and A.J. Jessurun (2001). "XML for flexibility and extensibility of design information models". In: *Proceedings of CAADRIA 2001*, Sydney, April 19-21, 2001.

van Leeuwen, J.P., Hendricx, A., and Fridqvist, S. (2001) "Towards Dynamic Information Modelling in Architectural Design." In: *Proceedings of the CIB-W78 International Conference IT in Construction in Africa 2001*, CSIR, Division of Building and Construction Technology, pp 19.1-14.

URL's

1. http://www.iai-international.org
2. http://www.designknowledge.info

Major Publications

The research results of this project have been reported with acknowledgement to its financial support mainly in the following publications:

van Leeuwen, Jos P. and S. Fridqvist. 2002. "On the Management of Sharing Design Knowledge." In: *Distributing Knowledge in Building*, Proceedings of CIB W78 2002, Aarhus, Danmark: June 12 – 14, 2002.

van Leeuwen, Jos P. 2002. "Knowledge Sharing for Collaborative Design." In: *Proceedings of 6th International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, Ellecom, The Netherlands, July 7-10, 2002.

van Leeuwen, Jos P. and S. Fridqvist. 2002. "Supporting Collaborative Design by Type Recognition and Knowledge Sharing." *Information Technology in Construction*, to appear in volume 8.

van Leeuwen, Jos P. and S. Fridqvist. 2002. "Design Knowledge Sharing through Internet Application." In: *Proceedings of the International Conference on Concurrent Engineering*, Cranfield, UK, July 27 – 31, 2002.

This report will be the basis for a final publication in the International Journal on *Automation in Construction*, to be submitted this year.