# An Ontological Decision Support System for the Design of Structural Simulation Models

Stefan Jaksch, Margit Rudy
Institute of Architectural Sciences
Vienna University of Technology
Karlsplatz 13/259-2, A-1040 Vienna
jaksch@iti.tuwien.ac.at, rudy@iti.tuwien.ac.at, http://www.iti.tuwien.ac.at

***Abstract:*** Decision support in conceptual design relies on the formalization of knowledge that is often approximate and ill-structured. We present a building information system that provides decision support in the early phases of architectural design, with a focus on the development of a building's load-bearing system. Common methods to represent relevant information in structural engineering are discussed and related to developments in knowledge representation and artificial intelligence. The implementation of various prototypes using an XML-based design ontology is described.

***Keywords:*** Knowledge representation, decision support, classification, ontology, information system, structural engineering

## 1 Introduction

Structural design is a creative process carried out by specialized engineers in close cooperation with architects and other planners. Numerous influences and sets of boundary conditions must be considered, leading to the solution and assessment of several alternate design options.

Information in the early design phases is largely approximate and not exactly defined. This is also reflected by information representation during conceptual design, which is still dominated by verbal descriptions, sketches and drawings, both digital and on paper.

Most often these representations cannot be interpreted automatically by computers; the semantics of the content require human interpretation. Due to lack of time, only few competing solutions can be thoroughly explored during the conceptual design phase.

The desire to reuse existing design knowledge (e.g., from previous design solutions) calls for new methods to record the decision-making process. With additional methods to retrieve and evaluate these processes, it should be possible to develop a greater variety of concepts and possibly gain more time for the investigation of innovative design ideas.

The notion of reusing different kinds of design solutions and ideas is widespread in architectural and structural design. Reusable design knowledge may consist of both standard design cases and specialty component solutions that have been successfully employed in a previous project.

As the use of object oriented methods and product modeling becomes more and more popular, today's information technology is maturing enough to adequately support engineers and architects in case based design tasks.

The application of object oriented techniques such as product modeling in CAD, however, requires clearly defined solutions for a design case, and all object parameters to be specified at a rather high level of detail. Consequently, common product models like IAI/IFC can be better employed during later design phases (see *2.2 Building Prod-*

*uct Modeling*), in which construction details are developed.

Moreover, the specialized object hierarchies of product models rarely resemble the terminologies of engineering practice. Although the objects and relations used in the product model are hidden behind advanced graphical user interfaces, problems can arise from this mismatch between the semantics of human and computer taxonomies.

Derived from a philosophical context, the term ontology is used in computer science to describe a system of domain concepts and their relations. An ontological approach can be used to model the semantics of a domain with logical statements suitable for computer representation – but in a way that is still close to the human understanding of that domain.

In our work, we examine the use of ontologies in order to model knowledge about the load-bearing behavior of buildings. With a focus on early design phases and conceptual planning, we try to develop methods to synthesize new load-bearing structures by reusing components of previously analyzed existing structures.

We describe decision support in an information system that enables the classification, storage and retrieval of available knowledge representations of a domain – in our case, structural engineering. Design knowledge is represented in a case database that contains both generic standard design solutions, and specialized design cases that have been constructed in reality. Each design step can be defined by relevant criteria that may be used to constrain the set of applicable design cases.

# 2 Objectives and Methodology

In the following sections, we characterize the type and quality of documents that are commonly used in architectural and structural design tasks. We briefly present methods for computer representation of design entities that are relevant in conceptual design stages. Combining some of these methods, we outline a strategy for synthesizing structural design models.

## 2.1 Data and Information in Architecture and Engineering

Design in the domain of structural engineering – as well as in architecture – requires information of many kinds (textual, graphic, geometric, topological, geographic, etc.) to describe different aspects of the designed building, such as its shape, extent, location, orientation, or topological relationships of spaces and components. Although much information is already available in the form of digital documents, the need for human interpretation of these documents remains (van Leeuwen 1999).

Computer-aided design (CAD) in everyday practice is still often used more as a drafting board replacement, rather than in a truly object-oriented manner. Even when the designer employs object-oriented techniques such as product modeling in CAD, the tools available require clearly defined solutions for a design case, since all object parameters have to be specified at a high level of detail.

Another family of standard software applications for structural engineering includes calculation tools and software for frame or finite element analysis. Though calculation results of such programs are of great importance for dimensioning the final design solution, tools for conceptual structural design that work with approximate values are few and far between.

A major step towards providing better structured data, as well as facilitating data exchange and interoperability, has been taken in recent years by the development and adoption of the Extensible Mark-up Language (XML), which is a text-based format for the structured description of data.[1] It is also possible to define the logical structure and (to some extent) the semantics of documents by using Document Type Declarations (DTD), or so called XML Schemas. Since it is a text-based language and, therefore, both human and machine readable, a longer life cycle can be expected for the information contained in XML documents than in application formats.

---

[1] More about XML: *http://w3.org/XML*

## 2.2 Building Product Modeling

An important effort to make architectural design data interpretable by computers has been made in the area of *product modeling*. Hereby not only the geometry of a designed object is modeled, but also all product properties that are relevant to any of the parties involved in the design process throughout the entire product life cycle. The modeled aspects may include, for example, material and other physical properties, or information about costs and labor.

Product models can facilitate the communication and exchange of design data between the different contributing parties in the design process. In combination with modern networking technologies, product modeling enables new levels of interoperability between the professions involved in the design and construction of buildings. The Industry Foundation Classes (IFC), a product model developed by the International Alliance for Interoperability (IAI), is at present supported by several important software manufacturers in the AEC sector.

Due to the high complexity of product models such as the IFC, as well as the fact that handling such complex data models containing hundreds of different object types can be error-prone, they still pose considerable problems in both implementation and application. Moreover, it is virtually impossible to define design objects as long as their details are not exactly known. This further complicates the application of standard product modeling during early phases of the design process, where the definition of design properties is still sketchy and imprecise.

In contrast to the described development of static product models like the IFC, research efforts to propagate the use of more dynamic data models are underway, especially in the area of conceptual design (cf., Fridqvist 2000, van Leeuwen 1999). These approaches establish less complicated object types in the form of application-oriented meta-classes, which enable the dynamic definition of domain concepts.

## 2.3 Classification and Ontologies

In order to retrieve design objects by specifying certain criteria, a knowledge representation system relies on taxonomic classification of its contents. A classification system with a common vocabulary is also important to improve communication between professionals, as well as to ensure that the technical terminology is applied consistently.

Common approaches to taxonomies are analogous to hierarchical library classification systems. Several national and international standards have been developed by this approach (e.g., ISO/TR 14177, ISO 12006-2, BSAB 96, SfB; cf., Ekholm 2002, Wright 1998). Since designers from distinct AEC domains (especially architects and structural engineers) have different views of the design object, their classification priorities may diverge substantially. Agreement on a common, strictly mono-hierarchical taxonomy is even likely to be counterproductive.

Therefore, from the designer's point of view, the need for classification also results from the necessity of a semantically well-defined vocabulary that can be used by all design participants. This calls for the development of an ontology that models different facets of design knowledge and information, while it semantically defines a common terminology for all.

The term ontology originates from philosophy and denotes "a systematic account on the nature and the organization of reality" (Simoff & Maher 1998). In the field of artificial intelligence, the concept of ontology stands for a system for representing domain concepts and their linguistic realizations by means of basic elements. With respect to design issues, “ontology defines the semantics of what is known about the design domain that the ontology covers” (Simoff & Maher 1998).

The combination of a design ontology with a simple, property-oriented product model appears to be a feasible approach for building the core of a knowledge-based system aimed at decision support.

## 2.4 Ontological Decision Support

The development work described in this paper intertwines various lines of research activity with the teaching activities at the author's university under the heading of an integrated project titled "*archistructura*: Media Development System for Building Science and Structural Design."[2]

The overall *archistructura* concept encompasses the following three main application areas, which are being developed in parallel:

- *design aids* – structural design support for architects and engineers,
- *study aids* – courseware and accompanying learning resources, as well as
- *buildings* – a database of documented design precedents with integrated case studies

(Rudy & Jaksch 2004).

Several hierarchical taxonomies for the classification of buildings and their load-bearing systems were developed in the context of the *archistructura* project. While these taxonomies served well for simple classification purposes in the growing building collection, their strict hierarchical structure did not prove to be flexible enough to support the projected extension of the system towards an explicit decision support application.

An important original objective of the *archistructura* initiative was to support product modeling. We stated earlier (cf., *2.2 Building Product Modeling*) that standard product models tend to define rather complex object hierarchies, and demand a variety of highly detailed data in order to establish design entities. Product models for conceptual design phases, in contrast, call for more flexibility and the ability to handle ill-defined and qualitative data. Such models are described as a "semantic model of conceptual entities and their relationships" (Stouffs & Krishnamurti 2001). It is also stated that these models "should support user-defined model object classes and reclassification of model object instances," thus being flexible enough to allow the classifying ontology system to be adapted during its development (Fridqvist 2000).

Based on these requirements, we started to develop a simple "meta-product model" to avoid the drawbacks of standard product models, namely the specification of a high level of detail and complexity. In order to define generalized representations of design objects, we established a main object type called *design item* (Fig. 1). These objects represent a "design" in the broadest sense of the word, ranging from a complete building design to the specification of such component details as the cross section of a beam.

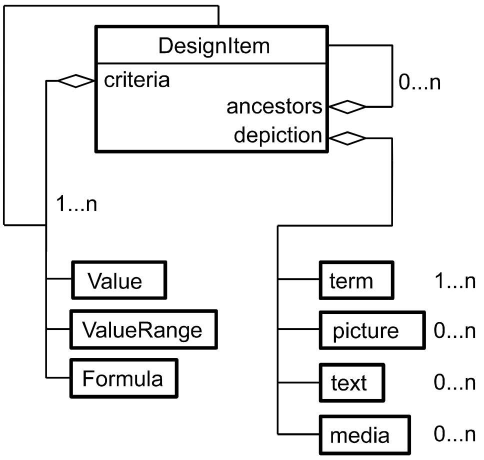


Fig. 1: Meta-objects used to define the ontology.

Each *design item* can link to one or more other items representing design tasks that may be designated to specify a design at a higher level of detail. Hierarchical structures can be outlined for design items that include inheritance mechanisms comparable to those in object-oriented programming languages. Thus it is possible to model relationships such as *system → load-bearing system → simple load-bearing system → beam → ...* (where the operator "→" can be read as "is a generalization of").

This means that the designer can draw a decision on a higher – more abstract – level (for instance, the decision "I will need a load-bearing system") and evolve the more detailed decisions in other design items, which describe the intended design more precisely. Alternatively, a comparatively detailed decision may be established at early stages of the design process (for instance, "I've decided that the main load-bearing

---

[2] *www.archistructura.net*

system is a beam”).

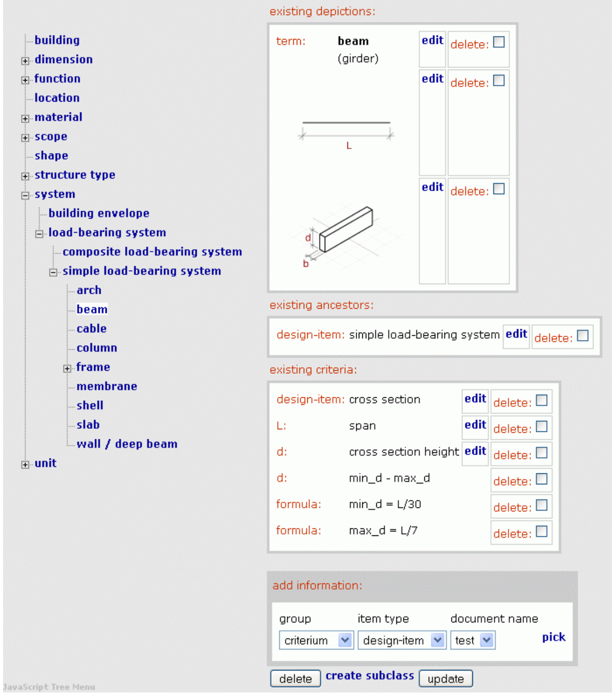
The described hierarchical relations between *design items* are realized by referencing the ancestors of an item. This allows the structuring of hierarchies with multiple inheritances, which is much more flexible than strict mono-hierarchies. A design item can thus have several parent objects (ancestors), e.g., a *truss* might be found as a child object of both *beam* and *latticed component*.

Design items are essentially classified by specifying their characteristic properties as *design criteria*. Such criteria can also be seen as the questions that need to be answered at the applicable stage of the design process as represented by the respective design item. The objects that can represent a design item's criteria can be either other items, or – especially at higher levels of detail – values, ranges of values and/or simple formulas for preliminary calculations (Fig. 2).

The transition from a design item to a criteria item represents a design decision that may require resolving a sub-design task. For example, the criteria *load-bearing system* within the design item of a building requires the definition of an instance of a load-bearing system item. The criteria representing the (sub-)design tasks of this item may involve such tasks as specifying the components and jointing of the load-bearing system, or describing component combinations in relation to the special context.

Depictions can be specified to render a given design item for the user. These can consist of descriptive terms, pictures, descriptive texts, or other media (e.g., CAD or multimedia files) that proved relevant in the course of design.

Although a predefined ontology covering the core domains of architecture and structural engineering is provided, this ontology can always be complemented, changed, and refined by its users (Fig. 2). Certain generic cases are included, representing standard applications and design solutions. New cases or adaptations of existing ones can be added to the database structure.



- building
- dimension
- function
- location
- material
- scope
- shape
- structure type
- system
  - building envelope
  - load-bearing system
    - composite load-bearing system
    - simple load-bearing system
      - arch
      - beam
      - cable
      - column
      - frame
      - membrane
      - shell
      - slab
      - wall / deep beam
- unit

existing depictions:

term: beam (girder)

existing ancestors:

design-item: simple load-bearing system

existing criteria:

design-item: cross section
L: span
d: cross section height
d: min_d - max_d
formula: min_d = L/30
formula: max_d = L/7

add information:

group: criterium
item type: design-item
document name: test

Fig. 2: Prototype environment for ontology development.

# 3 Implementation

The *archistructura* building information system was prototyped within an XML-based web development framework (*Apache Cocoon*[3]) in combination with *eXist*,[4] a native XML database. This framework has served well for a number of reasons, e.g.:

- It is possible to process XML data in software applications commonly used in structural design (such as spreadsheet or CAD programs).
- Processed data can be published in HTML via a web server, thus enabling network accessibility.
- Simple application logic can be comfortably implemented in XSLT, while more complex tasks can be solved using an object oriented language (Java).
- The use of a native XML database did not require the design of predefined database tables, thus facilitating possible changes in the data structure during development.
- The developed information has long-term persistence, thanks to the widespread use of XML-based technologies.

---

[3] *http://cocoon.apache.org*

[4] *http://exist-db.org*

In order to prove the overall concept, various prototype modules were developed for the desired integration in the conceived design-support environment. The main objectives of these modules include ontology development and enhancement, a management tool for design sessions, and tools for the informed selection and preliminary calculation of structural systems.

### 3.1 Ontology Editor

The classification system of the *archistructura* system was developed with a web-based tool using a tree structure to visualize the classification hierarchy. The web-based architecture proved especially useful when different people started working on the contents of various taxonomies, as the updated version is always available to every user in the network. On this technical basis, we started the implementation of a preliminary ontology editor, with XML data structures for design items and references between them. The functionality of that prototype included the definition of all basic properties of a design item (cf., Fig. 1):

- ancestor relationships for establishing the inheritance hierarchy,
- definition of necessary criteria (objects, values, ranges of values, and simple formulae),
- linkage to existing labels (internationalization), documents and images (depictions).

Using this tool, a first ontology for architectural design tasks was developed, with a focus on structural engineering and related topics.

As the ontology evolved, more and more entities had to be defined in increasing detail, and needs for special functions arose (e.g., to constrain criteria to discrete values or to define a required cardinality for a property). In order to avoid "reinventing the wheel," we decided to fall back on existing tools and export the draft version of our ontology to *Protégé 2000.*[5] This ontology editor was developed by Stanford Medical Informatics at the Stanford University School of Medicine and complies with existing semantic web standards (e.g., OWL and RDFS). Furthermore, it can be accessed by a Java API and thus be integrated in the existing *archistructura* implementation. This integration task is scheduled for the ongoing development of the structural design ontology.

### 3.2 Design Session Tool

The first application of our ontology is a simple decision support tool, where the user can perform typical steps in the design process with the aid of "rules of thumb" and a selection of subsidiary design tasks.

A web-based user interface provides a hierarchical tree containing the main entity classes of the ontology. After choosing a certain starting point for the design path (e.g., building function, location, load-bearing system), the user navigates the hierarchy of ontology entities. Possible design steps or decisions are determined and displayed depending on the context of a specific entity (Fig. 3). The user's choice is subsequently stored as an XML tree in the browser session. This "decision tree" represents alternate decision paths leading to different comparable solutions. Its root node can be seen as the finally chosen design for a building. Traversing the tree downward provides increasing levels of detail for the chosen entities for the different design aspects (criteria) and alternate sets of potential solutions.

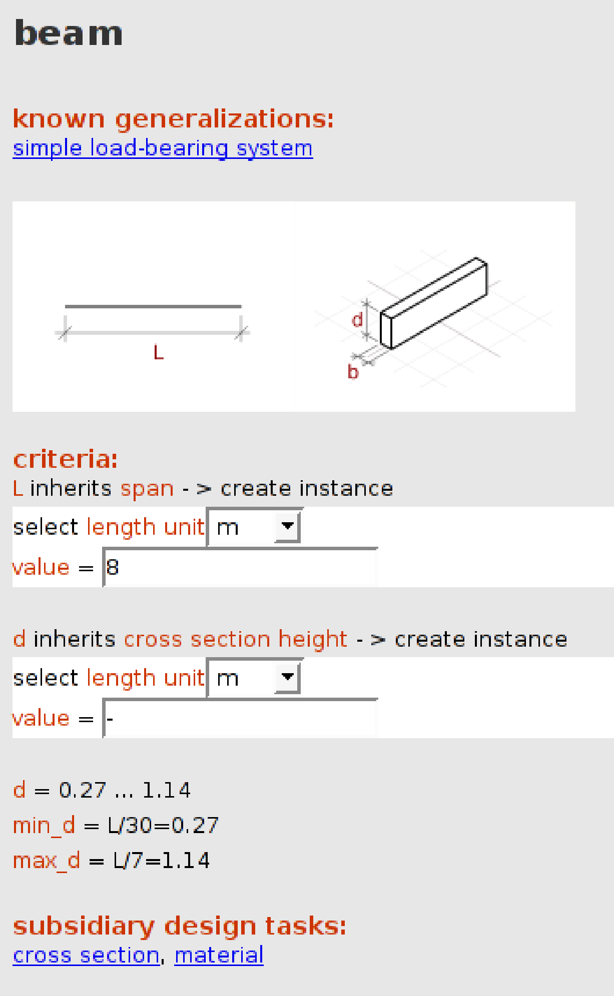
Relevant design items for each decision step are identified and evaluated on the basis of existing design session trees and decisions recorded in those trees for the same classes of design items. Therefore, the first prototypic design sessions consisted of the definition of generic design solutions for commonly used building categories (multistory buildings, roof structures, bridges, etc.). The most general of these paths postulates relations like "a building has a load-bearing system" or "the load-bearing system contains components and joints."

If the user is not satisfied with the proposed alternatives, the system can fall back on the solution space of an ancestor item, thus permitting unusual or

---

[5] *http://protege.stanford.edu/*

experimental decisions. In these cases, the user may be “warned” that he or she is about to leave the “territory of conventional solutions” for a certain task.



**beam**

**known generalizations:**
simple load-bearing system

**criteria:**
L inherits span - > create instance
select length unit m
value = 8

d inherits cross section height - > create instance
select length unit m
value = -

d = 0.27 ... 1.14
min_d = L/30=0.27
max_d = L/7=1.14

**subsidiary design tasks:**
cross section, material

Fig. 3: Session view of a simple design item (beam).

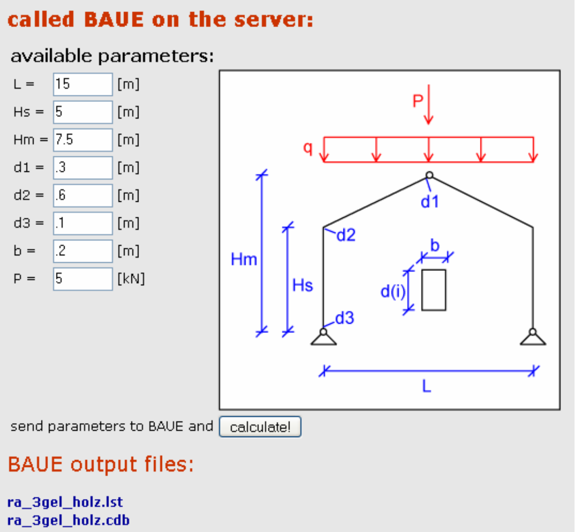
### 3.3 Calculation Web Service

As the determination of dimensions by simple rules of thumb is not always sufficient, additional tools were implemented to prepare the selected structural components for calculation with frame analysis software.

We developed templates for a range of common structural systems (beams, frames, arches, etc.). These templates are used to dynamically generate input files for a structural analysis program. Standard values are preset for parameters not yet explicitly defined at the current stage of the design project (e.g., standard load cases based on the intended function of the component).

The dynamically created file can be sent to an application server that provides the calculation service for simulating the design's structural behavior (Fig. 4). Calculation results are returned to the client and can be visualized three-dimensionally (Fig. 5).

Recent developments include mechanisms for combining such component models in a manner that enables the generation of more complex simulation models for analyzing the behavior of an entire building structure.



**called BAUE on the server:**

available parameters:

L = 15 [m]
Hs = 5 [m]
Hm = 7.5 [m]
d1 = .3 [m]
d2 = .6 [m]
d3 = .1 [m]
b = .2 [m]
P = 5 [kN]

send parameters to BAUE and calculate!

**BAUE output files:**

ra_3gel_holz.lst
ra_3gel_holz.cdb

Fig. 4: Input values for a pin-jointed frame and output of the calculation web service

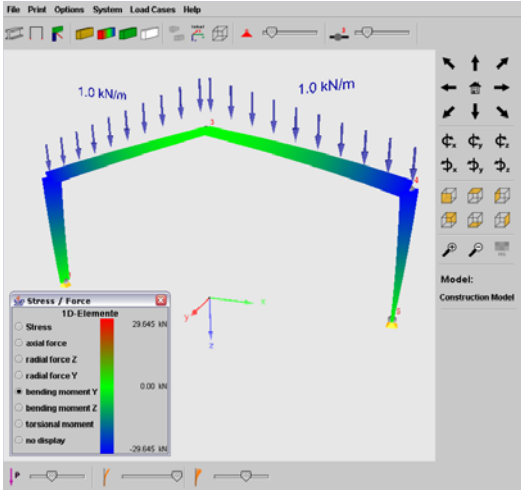


Fig. 5: Pin-jointed frame visualization.

## 4 Summary and Prospects

Applying knowledge based techniques is especially difficult in such complex domains as conceptual structural engineering, where many quantitative parameters are imprecisely defined and the domain terminology may be used inconsistently. The experimental prototypes presented in this work address some of the problems of this area; further testing and integration within the context of a complex building information system is projected.

In the presented approach, we seek to combine knowledge about the load-bearing behavior of both generic and exemplary designs with aspects of other architectural functions. First results of modeling the relations of load-bearing structures and architectural spaces are promising and allow us to conclude that the proposed semantic descriptions are useful in supporting conceptual building design. The combined use of tools for estimating design dimensions enables a qualitative evaluation of the feasibility of a developed conceptual design.

The implemented prototypes shall be further developed in the future with an aim to overcome current limitations. Especially the rather static definition of parameters in the templates for preliminary calculation ought to be fully integrated into our significantly more flexible ontology management system.

The *archistructura* building information system is being developed mainly in an educational context (Rudy & Jaksch 2004). The described ontology shall be enhanced to define basic knowledge of structural design such that this knowledge can be made accessible to students via a web application that implements the prototype session manager and decision support application. The tool will subsequently serve as a design aid for finding structural system dimensions during student design projects.

The close dialog with students as prototype users and their project evaluation has already proven very useful in related educational projects within the *archistructura* context. Future developments include the expansion of the design ontology to other domains of architecture, the application of the decision support system to analyze existing buildings, and the integration of parameterized structural components that can be further processed by analytical calculation tools.

***References:***

Ekholm, A. (2002). "Principles for classification of properties of construction objects." In Agger K., Christiansson P. and Howard R., (Eds.): *Distributing Knowledge in Building - CIB W78 Conference 2002*

Fridqvist, S. (2000). *Property-oriented Information Systems for Design*. Ph.D. thesis, Department of Construction and Architecture, Lund Institute of Technology, ISBN: 91-7874-062-2

Rudy, M., & Jaksch, S. (2004). "Building Science by Design: Developing a Building Information System and Teaching Architecture with Analytical Case Studies." *ED-MEDIA 2004: World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Association for the Advancement of Computing in Education, Norfolk (VA), USA, 4815-4820.

Simoff, S.J., & Maher, M.L. (1998). "Ontology-based multimedia data mining for design information retrieval." *Proceedings of the ACSE Computing Congress*, Cambridge (MA), USA, 310-320.

Stouffs, R., & Krishnamurti, R. (2001). "Standardization: A Critical View." *Proceedings of the CIB-W78 International Conference: IT in Construction in Africa 2001*, CSIR, Pretoria, South Africa, 42.1-42.10.

van Leeuwen, J.P (1999). *Modelling Architectural Design Information by Features*. Ph.D. Thesis, Eindhoven University of Technology/Faculteit Bouwkunde, Bouwstenen 54, Eindhoven, Netherlands.

Wright, T. (1998) "Terminology for object modelling of construction information." *A report on some issues for ISO/TC59/SC13.* Construction Information Systems Australia Pty Ltd

*Acknowledgments:* Concept and tool development for the *archistructura* project were supported in part by a grant from the Austrian Science Fund (research project titled *Building Information Systems as Interactive Design Support*).