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, 1040 Vienna
AUSTRIA
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 have to be considered, resulting in the development and evaluation of several design solutions and variants of these solutions.

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 variants can be thoroughly explored during the conceptual design phase.

Since it is desirable to reuse existing design knowledge (e.g., from previous design solutions), the development of methods to record the decision-making process is required. 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 be comprised 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 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 Product 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. To achieve a scalable level of detail for developing new structural designs, we have also been investigating potential means of converting these conceptual models to standardized product model formats so that they may be reused in existing software applications.

A decision support system that provides information about generic design cases and built structures could aid the process of early structural design and facilitate the creation of alternate solutions.

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 base that contains both generic standard design solutions, and specialized design cases that have been constructed in reality. Each design step can be defined by criteria that may be used to constrain the set of applicable design cases.

2 Objectives and Methodology

2.1 Data and Information in Architecture and Engineering

Data can be defined as the "record of properties of arbitrary entities, (...) not bound to any specific system" or, regarding criteria, as "accuracy, precision, semantics and so forth (...) Data becomes information as soon as it is found to be relevant by any operational system." [1].

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, there is still the need for human interpretation of these documents [2].

Available design information can be classified according to its data quality [3]:

- structured data might be represented by attribute-value pairs, relational tables, and/or object-oriented structures;
- weakly structured data in the form of word-processing and spreadsheet files, or drafting-oriented CAD data;
- raw data (e.g., raster images, graphics, audio or video files).

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. Thus the acronym "CAD" might be more appropriately translated as *computer aided drafting* rather than *computer aided design*. 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 rather high level of detail.

Another family of classic software applications for structural engineering includes calculation tools and software for frame or finite element analysis. The calculation results of these programs are of great importance for dimensioning the final design solution. Nevertheless, 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 and, therefore, both human and machine readable language, a longer life cycle can be expected for the information contained in XML documents than in application formats.

Many recent versions of software applications used in building design support the export of XML data and can thus be used to convert older files. XML mappings of product model schemata (cf., following sections) either already exist or are currently being developed.

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 properties of the product that are relevant to all 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 parties involved 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 International Alliance for Interoperability (IAI) was founded in 1995 to improve software interoperability in the AEC/FM industry (Architecture, Engineering, and Construction / Facility Management). The developed product model, the Industry Foundation Classes (IFC), is nowadays supported by several important software manufacturers in the sector.

Due to the high complexity of product models like the IFC and the fact that handling such complex

---

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

data models containing hundreds of different object types can be error-prone, there are still substantial difficulties in both implementing and applying them. Furthermore, it is virtually impossible to define design objects as long as their details are not exactly known. This complicates the application of classical product modeling during early phases of the design process, where the definition of design properties is still sketchy and imprecise.

Current implementations of product modeling concepts appear to focus mainly on data relevant to business processes, e.g., for facility management or construction controlling. Also well supported is the detailed description of elaborated designs, though not their conceptual development. Although structural design data is included in some models, the use of product model data is not yet common in structural engineering.

In contrast to the described implementation and application efforts of static product models like the IFC, there are research efforts to propagate the use of more dynamic data models, especially in the area of conceptual design [2],[4]. These approaches establish less complicated object types in the form of application-oriented meta-classes, which enable 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. Classification and 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 related to hierarchical library classification systems. Several national and international standards have been developed according to this approach (e.g., ISO/TR 14177, ISO 12006-2, BSAB 96, SfB) [5],[6]. Since designers from distinct AEC domains (especially architects and structural engineers) have different views of the design object, their classification priorites may diverge considerable. Agreement on a common, strictly mono-hierarchical classification system 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 design knowledge and information, while it semantically defines a common terminology.

The term ontology originates from philosophy and denotes "a systematic account on the nature and the organization of reality" [3]. 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” [3].
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 that incorporates 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

[4].

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 applications.

An important original aim of the *archistructura* initiative was to support product modeling. We stated earlier (cf., *2.2 Building Product Modeling*) that classical 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” [8]. It is also stated that these models “should support user-defined model object classes and re-classification of model object instances,” thus being flexible enough to allow changes of the classifying ontology system during its development [4].

Based on these requirements, we started to develop a simple “meta-product model” to avoid the drawbacks of standard product models, namely the

---

[2] *www.iti.tuwien.ac.at/projekte/ars/index_en.htm*

specification of a high level of detail and complexity.

In order to establish fundamental 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.

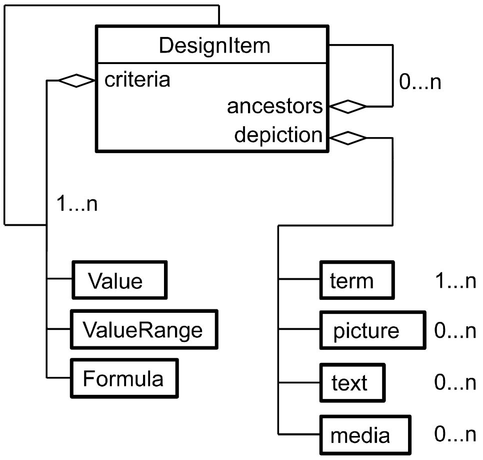


Fig. 1: Meta-objects used to define the ontology.

Each *design item* can link to one or more other items representing design tasks that may be specified as defining 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 → girder → ...* (where the operator "→" can be read as "is a generalization of").

This means that the designer can draw a decision on a higher – rather 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 example, "I choose a beam as my main load-bearing system").

The described hierarchical relations between *design items* is 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.

The transition from one design item to another, reasoned through interpretation of criteria by the user, represents a design decision that may require resolving a sub-design task.

A simple inference mechanism was included which allows chains of criteria to be evaluated with respect to purpose. Constraints and obligatory values (e.g., for simulation) are checked by traversing an item's ancestors and other criteria.

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). With this mechanism, any files that were relevant in a design process can be archived and kept available to guide subsequent decisions.

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. 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.

Evaluation of constraints, feature combinations, and other aspects of the given ontology is left to the implemented application logic. Thus it is possible to react differently to specially defined criteria that are applied repeatedly in a variety of design contexts (such as, for examples, units or other dimensional properties).

3 Implementation

The *archistructura* building information system was developed within an XML-based web development framework (*Apache Cocoon*[3]) in combination with *eXist*[4], a native XML database. Among the benefits of using an XML-based framework are the long-term persistence of the developed information and the flexibility of the data model, which proved ideal in prototyping.

In order to prove the overall concept, various prototype modules were developed for the desired integration of decision support for structural design purposes. The main objectives of these modules include ontology development and enhancement, a

---

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

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

management tool for design sessions, and tools for definition and preliminary calculation of structural systems.

3.1 Ontology Editor

The classification system of the *architectructura* system was developed in 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 the classification, as the updated version is always available to every user via network connection. On this technical basis we started the implementation of a preliminary ontology editor, based upon XML data structures for design items and references between them. The functionality of that prototype includes the definition of all basic properties of a design item (cf., Fig. 1).

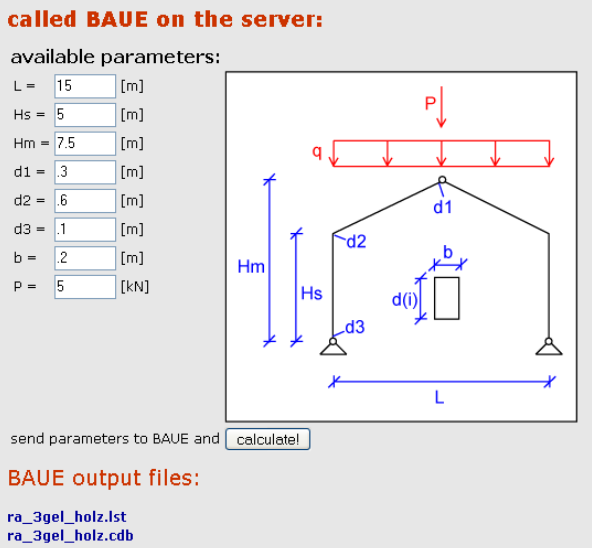
Using this tool, a first ontology for architectural design tasks was developed, with a focus on structural engineering and related topics.

As the ontology evolved, more and more entities had to be defined on a detailed level and the need 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 import 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 *architectructura* 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, supported by rules of thumb and a selection of subsidiary design tasks.

After choosing a certain starting point for the design process (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. The user's choice is subsequently stored in the browser session. Thus a “decision tree” is constructed that represents the decision path through relevant variants.



called BAUE on the server:

available parameters:

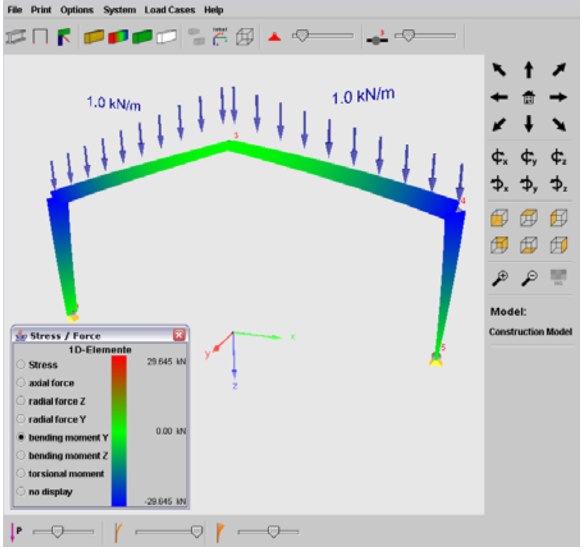
L = 15 [m]
Hs = 5 [m]
Hm = 7.5 [m]
d1 = .3 [m]
d2 = .6 [m]
d3 = .1 [m]
b = .2 [m]
P = 5 [kN]

send parameters to BAUE and

BAUE output files:

ra_3gel_holz.lst
ra_3gel_holz.cdb

Fig. 2: Input values for a pin-jointed frame and output of the calculation webservice



Stress / Force

1D-Elemente

○ Stress
○ axial force
○ radial force Z
○ radial force Y
● bending moment Y
○ bending moment Z
○ torsional moment
○ no display

29.645 kN
0.00 kN
-29.645 kN

Fig. 3: Pin-jointed frame visualization.

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 frame analysis program in a common product model format [9]. 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 a service for the simulation of the component's structural behavior (Fig. 2).

---

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

Calculation results are returned to the client and can be visualized three-dimensionally (Fig. 3).

Recent developments include mechanisms for combining such component models in a manner that enables the generation of more complex simulation models for analyzing the structural behavior of a whole building (Fig. 4).

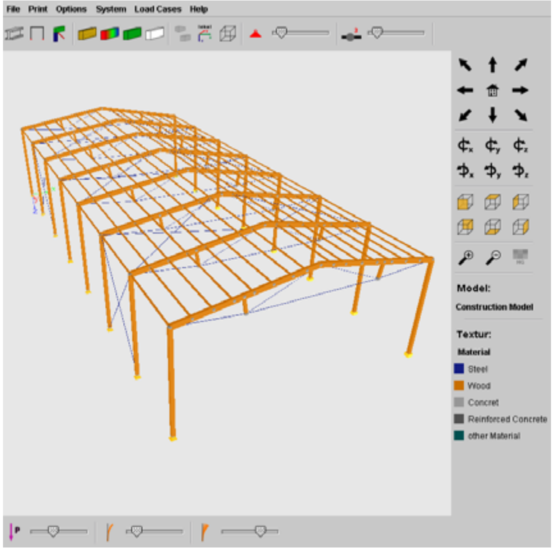


Fig. 4: Combination of component systems to form a complex load-bearing structure

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.

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 system. 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:

[1] Schatten, A. (2003) Sustainable Web-Based Organisation of Project-Related Information and Knowledge. (Ph.D. Thesis), Institute of Software Technology and Interactive Systems, Vienna University of Technology.

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

[3] 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.

[4] Fridqvist, S. (2000) *Property-oriented Information Systems for Design*. Ph.D. thesis, Department of Construction and Architecture, Lund Institute of Technology, http://www.caad.lth.se, ISBN: 91-7874-062-2.

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

[6] 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.

[7] 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.

[8] 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.

[9] DSTV (2000) Standardbeschreibung Produktschnittstelle Stahlbau. Deutscher Stahlbau-Verband (German Steel Construction Association), http://www.deutscherstahlbau.de.

Acknowledgments: The *archistructura* project was supported in part by a grant from the Austrian Science Fund (research project titled *Building Information Systems as Interactive Design Support*).