SAM: A MULTIMEDIA CASE LIBRARY OF STRUCTURAL DESIGNS

MARY LOU MAHER
Key Centre of Design Computing
Department of Architectural and Design Science
University of Sydney
mary@arch.usyd.edu.au

Abstract. Recent developments in multimedia and case-based reasoning provide the basis for developing teaching aids for architecture students that present technology and science learning materials as design cases. Case-based reasoning tools can provide assistance in the identification of a relevant design case and the modification of a case for the current design problem. We have developed multimedia library of buildings to support a case-based reasoning approach to teaching structural design . The design cases are linked through a network of concepts that follow a specific learning area, for example, the structural design of tall buildings is linked through the concept of lateral load resistance. The multimedia environment provides an active learning tool that the student uses to generate design solutions.

1. Introduction

Recent developments in multimedia and case-based reasoning provide the basis for developing teaching aids for architecture students that present technology and science learning materials as design cases. Case-based reasoning tools can provide assistance in the identification of a relevant design case and the modification of a case for the current design problem. Case-based reasoning is an approach to problem solving that uses a representation of previously solved problems as the basis for solving new problems. The approach incorporates an explicit representation of previously solved problems and identifies techniques for reminding and modifying this representation for the new problem. This approach changes a database of previous designs into an active learning experience rather than a passive resource.

We have developed a case-based reasoning tool for teaching structural design called SAM (for Structures And Materials). This approach begins to bridge the gap between the way students are taught architectural design and architectural science. Through interaction with SAM, the set of design cases grows to include the student’s own designs. Students are given a methodology (case-based reasoning) and a tool (SAM) for incorporating the knowledge of structural systems in the context of a design project. Learning is improved through the student ownership of the design cases and their own designs in an interactive medium. Students are taught architectural science by using design problems as vehicles and are given a tool that allows them to direct their own learning in the context of a design problem.

Studio-based learning forms the core of architectural education and architectural design is considered the most important of all subjects. In contrast, learning architectural science subjects is based on lectures and problem sets. Students generally do not respond to lecture-based courses because of the difference to the design studio style of teaching and, as a result, the learning outcomes are not as effective. Teaching these subjects has two aspects: learning the subject matter (usually lecture-based presentation) and learning to use this knowledge within the design context (usually through assignments). Case-based reasoning is a way of making experience explicit and reusing the relevant experience in learning to solve a new problem. Using case-based design as an approach to teaching structural systems promises to improve learning outcomes by relating the teaching methods in architectural science to those used in design studio.

2. Design as case-based reasoning

Case-based reasoning is an approach to problem solving that falls under the more general category of reasoning by analogy. Analogical reasoning is based on the idea that problems or experiences outside the one we are currently dealing with may provide some insight or assistance. Through analogy, we may be reminded of the way a braced frame system worked in a specific building when designing a structural system for a new building. Analogy is a way of resolving something that has not been encountered before by associating it with something that has.

The development of formal models of analogical reasoning has been studied by researchers in Artificial Intelligence. An early model of analogical reasoning for problem solving is reported in Carbonell (1981). The development of this model has led to new approaches to machine learning (Carbonell, 1983), and to the distinction between derivational analogy and transformational analogy (Carbonell, 1983, 1986). Considering analogy from the perspective of memory and reminding has developed somewhat independently of analogical reasoning and has led to the concept of memory organisation (Schank, 1982) as a guideline for computer representations. This work has been extended, applied, and developed into an area of AI-based problem solving called *case-based reasoning* (Riesbeck and Schank, 1989; Kolodner, 1994). Case-based reasoning has continued as a research area within AI but has also been applied and used in real-world situations.

Case-based reasoning, as a design process, is illustrated in Figure 1. A case library provides several examples of designs and the basis for finding relevant designs to a new design problem. A new design problem provides some information that serves as the basis for recalling one or more design cases. A selected design case can then be adapted to be a new design. The resulting new design can be added to the case library, allowing the library to grow with use.

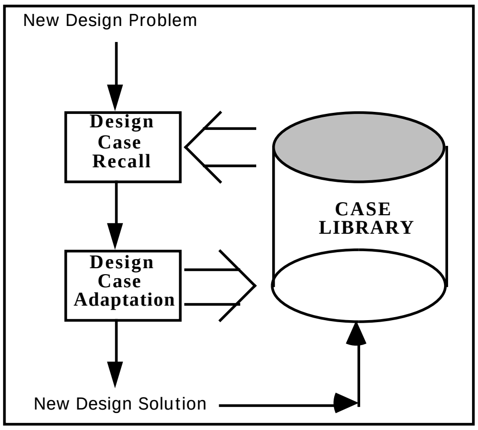

Figure 1. Design process using case-based reasoning

The applications of case-based reasoning to design have ranged from largely informal domains, such as meal planning, to analytical domains, such as the design of mechanical devices (for publications that describe several applications of case-base reasoning in design see Pu (1994), Maher et. al (1995) and Maher and Pu (1997)). Each design application, eg structural design of buildings, brings a new set of considerations to the use of case-based reasoning in design. The application of case-based reasoning to design has resulted in a number of implementations that resolve the theoretical issues raised by the paradigm itself, ie case memory organisation.

Here, we consider the issues raised by the need to organise learning material within a multimedia case library of structural designs. Specifically, we consider:

1. the need to represent and manage *complex design* cases,
2. the need to formalise a typically *informal body of knowledge* or experiences.

Design in any domain usually involves the development and understanding of complex systems. The complex representations needed to adequately capture a design case have introduced challenges to CBR systems. The CBR paradigm assumes that there is a concept of "a case", but in most design domains this concept is not simply "a case" but a complex set of experiences and decisions resulting in a complex system. Three approaches to addressing complexity are:

1. a case is a *hierarchy of concepts***,** or subcases
2. a case is represented by *different views*
3. a case is presented as *multimedia*

A general approach to addressing domain complexity is the representation and reuse of parts of cases, typically organised as hierarchies of "subcases." This supports case-based reasoning because subdividing designs in this way allows reasoning to focus only on the relevant parts of a design. By processing only some of the knowledge associated with a case, reasoning can become more efficient. The development of a case-base that has a hierarchical structure usually requires defining a typical decomposition of a design experience.

The use of different views of a design case recognises that a design can be understood from different perspectives. In this approach, a single, complex design project is represented as multiple cases. The use of multimedia can make it easier to understand complex systems - icons, images, sketches, etc. can highlight and illustrate corresponding text or tabular information.

The lack of formal knowledge in design affects both the ability to define a formal and consistent representation of design cases and the role of adaptation as a human-centred activity or an automated process. The development of CBR for design domains in which there is little formal or theoretical knowledge has been pursued by either formalising knowledge that previously was not formalised, eg by using an object-oriented representation, or by identifying a representation of the design cases to support human reasoning rather than automated reasoning, eg by creating a multimedia presentation.

Resolving the issue of "what is in a design case?" is done in many different ways. Contrary to the initial observation that case acquisition should be straightforward, most design stories told by designers or found in design documents are not easily formed into cases that can be indexed and classified for reuse. A systematic approach is needed to identify a uniform representation and to parse design stories into these formats.

3. Representation of Structural Design Cases

The representation of design cases in SAM follows the structural design principals that are taught in the Structures and Materials course. The overall organisation of design information falls into three categories:

1. project information,
2. functional decomposition of the structural design, and

3. structural system types.

These three categories are reflected in the navigation aids provided on each page as links to other parts of the case description. Figure 2 shows a typical presentation of the case description as the front page of the Grosvenor Place building. The front page is described further below. The functional decomposition of the structure is reflected in the links to the vertical load system, the lateral load system, and the footings. The structural systems types are shown a links along the third row, in Grosvenor Place the primary structural systems used are slab, columns, and core.

Figure 2. Front page of Grosvenor Place

The project information is presented on the "front page" of each case, providing a tabular overview of the project. The information includes:

Project data - this describes the general information about the building including the people involved and the overall geometry.

Design requirements - this describes the context in which the building was designed considering the city planning issues, the architect’s decisions about the overall shape and function of the building, the specifications for load and foundation conditions, etc. This information represents the considerations in the transition from the design brief to the structural design requirements.

Design alternatives and solution - this describes the different structural solutions considered by the engineers and the reasons for selecting the one implemented.

Justification - this describes the reasons for many of decisions that resulted in the final structural system.

References - acknowledges the sources of information.

The functional decomposition of the structural design is reflected in the set of "pages" that refer to the vertical, lateral, and footings systems. Each "page" further decomposes the information related to each functional system, for example, the vertical load systems can be further decomposed into gravitational and uplift load systems. Structural efficiency issues are dealt with at a number of levels - load transfer strategy, load paths, and structural actions, as illustrated for the vertical load resisting system in the Sydney International Aquatic Centre in Figure 3.

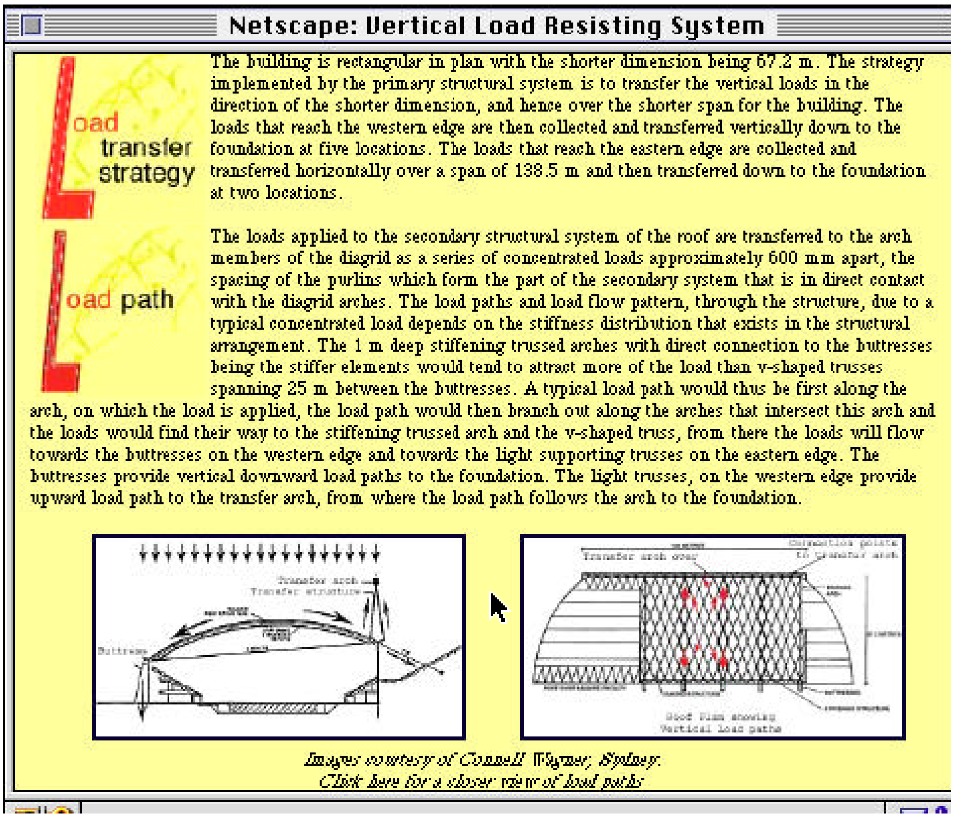

Figure 3. Lateral load resisting system in the Sydney International Aquatic Centre

The structural systems are also described as separate “pages”. Each structural system is introduced generically as a type of structure, and then described in the context of the geometry and load conditions of the building case study. Each structural system is presented in terms of how it contributes to the functional decomposition. For example, the core system in Grosvenor Place is described as providing both lateral and vertical load resistance.

4. Using the case library

Students can use the case library in either a learning mode or a problem solving mode. In the learning mode the students can interact with the system to learn about structural design issues - such as assembling structural systems and making design decisions, or structural principles - mainly those relating to structural efficiency. In the problem solving mode, students can use the system to assist with their design projects.

The case library is implemented in an application called SAM (Structures And Materials). The cases are stored as a set of files written in HTML (Hypertext Markup Language) and the cases are browsed a WWW browser. The basic uses of the case library are: search or browse the case library; and modify and add a case to the case library, as described below.

Searching. There are three indexing schemes for searching for design cases in SAM:

Search by name of building: This option allows the user to view a list of all the buildings in SAM. The buildings are sorted according to two types of buildings: wide span and tall buildings. For each type, a list of names of buildings is given, where each name is linked to the first page of the case description of the building. This approach is useful when the user is interested in a specific building and recognises the name of a building.

Keyword search: This option allows the user to complete a form in which a structural design can be described in terms of the architect, engineer, types of structural system, structural materials, etc. The application then returns a list of all buildings that satisfy a partial match with the specified keywords. The buildings are ranked according to how close the match is. This approach is useful when the user is not sure which building is relevant, but knows something about the new case study or design he/she is working on.

String search: This option allows the user to specify a list of words that may be used in a case description. The application returns a list of cases that include the specified strings in the case description. This approach is similar to the search engines on the WWW and is useful when the keyword search did not result in a useful set of cases.

Browsing. A case is made up of several "pages". Each page has a title indicating a general heading for the content of the page, and a photograph illustrating the content. Each page has a set of navigation aids: a large scroll provides a direct link to the contents of the current page, and several smaller scrolls provide direct links to the other pages that describe this case. A button provides access back to the home page of SAM which allows the user to find a different case.

Modifying. The modification of a case is done by saving the contents of a case into new files and then editing the files to remove irrelevant information and add new information related to the new case study or design. This can be done using the functions in the WWW browser to save HTML and image files, followed by using an editor to change the contents of the files. SAM does not provide direct assistance in "case adaptation" since this is an important learning effort for the student.

Adding. Adding a new case to SAM can be done by selecting the "Build SAM" button on SAM's home page. This button leads to a form that is identical to the keyword search form linked to the "Find SAM" button. The user specifies the keywords for the new case plus the URL of the first page of the case. The next time a user searches for a case, the new case will be included.

Conclusions

This approach to teaching architecture students is effective because it relates architectural science to the way in which students learn to design, because it gives students ownership of their learning, and because case-based reasoning enhances links between short term memory and long term memory. Through the identification of a case as related to the current problem, the students learn by linking long term memory (cases) with short term memory (the new problem) - this enables deep learning.

This project integrates successfully into the total learning process because it brings the teaching of architectural science towards the way in which architectural design is taught with a major difference being that the students have access to the cases whenever they want, rather than only through the studio instructor or the books in the library. This project also provides an environment in which the student learning occurs in the same medium as the student assessment. The students use the case-based reasoning tool as a resource when developing the design solutions and use the same medium to submit their assignment. Assignments are submitted as a disk with their newly designed case in a multimedia presentation. Through interaction with this tool, the set of design cases grows to include the student's own designs. Learning is improved by giving students a method and a tool for incorporating the knowledge of architectural science material in the context of a design project.

Acknowledgments. This project was funded by a CAUT grant. The co-investigators of the project, Terry Purcell and David Gunaratnum, collaborated with the author on many of the ideas presented in this paper. David Gunaratnum teaches the Structures and Materials course in which SAM is used and has provided his expertise in shaping the content and organisation of the cases. The presentation of the cases as multimedia has benefited from the hard work and creativity of Diella Bolzan and Fiona Kerr. The organisation of the cases in a database and the search engines have been inspired and implemented by Phil Tomlinson and Simeon Simoff. The details of the cases themselves are courtesy of the engineers and architects that work on the projects - the individuals are named in the case library in the reference section for each case (http://www.arch.usyd.edu.u/kcdc/caut/) .

References

Carbonell, J. G.: 1981, A computational model of analogical problem solving. Proceedings of the Seventh International Joint Conference on Artificial Intelligence 1: 147-152.

Carbonell, J. G.: 1983, Learning by analogy: Formulating and generalising plans from past experience. *Machine Learning: Volume 1. An Artificial Intelligence Approach.* Palo Alto, CA:Tioga, pp. 137-161.

Carbonell, J. G.: 1986, Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. *Machine Learning: Volume 2. An Artificial Intelligence Approach*. San Mateo, CA: Morgan Kaufmann, pp. 371-391.

Kolodner, J. L.: 1993, *Case-Based Reasoning,* Morgan Kaufmann, New York.

Kolodner, J. L., and Riesbeck, C. K. (1986). *Experience, Memory, and Reasoning.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Maher, M. L., Balachandran, B., and Zhang, D. M.: 1995, *Case-Based Reasoning in Design,* Lawrence Erlbaum Associates, Hillsdale, NJ.

Maher, M.L. and Pu, P. (eds.): 1997, *Issues and Applications of Case-Based Reasoning in Design,* Lawrence Erlbaum Associates, Hillsdale, NJ.

Schank, R. C.: 1982, *Dynamic Memory: A Theory of Learning in Computers and People,* Cambridge University Press, Cambridge.

Riesbeck, C. and Schank, R. C.: 1989, *Inside Case-based Reasoning,* Lawrence Erlbaum Associates, Hillsdale, NJ.