eCAADe: An Educational Commendation Mechanism for the Adaptive Semantic Web to Use in the Architectural Design Environment

Ji-Hyun Lee[1], Shu-Feng Pan[2]
Graduate School of Computational Design, National Yunlin University of Science & Technology, Taiwan
[1]http://jihyun.gcd.yuntech.edu.tw/, [2]http://mail.gcd.yuntech.edu.tw/~dicky/

The "Case Studies in Architecture" phase of a course in architectural design includes in-depth comparisons and analyses of architectural precedents. However, with the large number of cases now readily available, Web searching and navigation is a time-consuming, low precision activity. In the work described in this paper, we built eCAADe, an educational commendation system for the adaptive semantic Web to allow students to query and retrieve semantically for architectural cases during the case study phase of an architectural design process. In our suggested system, we built a Semantic Web for design knowledge representation to make query and retrieval efficient. We also applied a hybrid recommendation mechanism, which is combining both content-based filtering and collaborative filtering to help for students to find relevant cases more efficient and precise with their preferences. We illustrate our concepts with several concrete examples.

Keywords: *Recommendation mechanism; adaptive; semantic Web; architectural cases; ontology.*

Introduction

The "Case Studies in Architecture" phase of a course in architectural design includes in-depth comparisons and analyses of architectural precedents. Case studies provide students with detailed information about the conditions and problems of various past architectural precedents. The task of a case study consists of data collection and data analysis. There are several online case-based systems such as DYNAMO (Heylighen, 2000) and archINForm[1], which provide a search and retrieval engine for design cases to use by students and designers.

However, with the large number of cases now readily available, Web searching and navigation is a time-consuming, low precision activity. Therefore, we require a recommendation mechanism that is adaptive on the basis of the user's various needs and of explicit and implicit user feedback during the task of finding relevant information in complicated structures of websites. Recommendation needs effective techniques for information filtering and information retrieval. General keyword-based search engines present several problems when used for this purpose. (1) The search results may show high

[1] www.archinform.net

recall, but low precision; (2) The results may show low or no recall: users often get no answer to their request, or relevant pages are not retrieved; (3) The results may be highly sensitive to vocabulary; (4) The results may be single Web page (Antoniou and van Harmelen, 2004). Because of this, we need to use machine-processable Web content to represent domain knowledge. The Semantic Web is an initiative that is aimed at improving the situation in relation to the World Wide Web. By using structured semantic descriptions, users can index and retrieve information with more precision.

In the work described in this paper, we built eCAADe, an Educational Commendation mechanism for the Adaptive semantic Web to use in the Architectural Design Environment, to allow students to query and retrieve semantically for architectural cases during the case study phase of an architectural design process. We illustrate our concepts with several concrete examples.

Architectural Case Studies and Case Bases

Case studies are an essential pre-requisite to any architectural design process. Students are introduced to a well documented approach such as pictures, drawings and text for investigation and understanding of remarkable buildings and are requested to collect architectural cases to apply the approach to analyze the cases (Zimring et al, 1996). A case base is a collection of specific cases stored as complete patterns of experiences from that past, and labeled by a set of characteristic features serving as indices (Heylighen, 2000). A case-based system should store the design context of cases in terms of features, graphics, and pieces of text that enable users to associate elements with the contexts or situations of designs or to reason about those contexts or situations, and the system then helps users to retrieve the most appropriate cases (Lin and Chiu, 2003).

With the advent of the World Wide Web, the Web has evolved into a user-interactive medium capable of delivering on demand information in high speed. We, therefore, benefit to build a case base on the Web as a learning platform for students to retrieve architectural cases.

Ontologies and Semantic Web Technologies for Knowledge Representation

Tim Berners-Lee, the inventor of the World Wide Web, coined the name "Semantic Web" for a web of machine-readable information whose meaning is well-defined by standards: it absolutely needs an interoperable infrastructure of a kind that only global standard protocols can provide (Fensel et al., 2005). Figure 1 shows the main layers of the design and vision of the Semantic Web. From bottom to top, it contains the XML, RDF, Ontology, Logic, Proof and Trust layers (Antoniou and van Harmelen, 2004).

In 1993, Gruber gave the definition that an "*ontology is a formal, explicit specification of a shared conceptualization*". In general, ontology consists of a finite list of terms and the relationships between these terms. The terms denote important concepts (classes of objects) of the domain (Antoniou and van Harmelen, 2004). Ontologies were developed in the field of artificial intelligence to facilitate knowledge sharing and reuse. More recently, the notion of ontology has also become widespread in fields such as intelligent information integration, cooperative information systems, information retrieval, electronic commerce, and knowledge management (Fensel et al, 2005). Guarino (1998) proposed four different kinds of ontologies, according to their level of de-

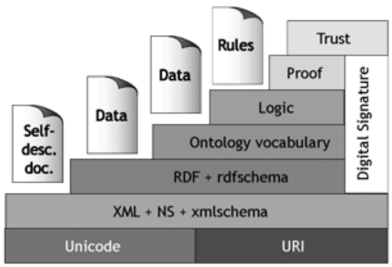

Figure 1
A layered approach to the Semantic Web (Antoniou and van Harmelen, 2004).

pendence on a particular task or point of view. One of them is the class of *domain ontologies*, which describe the vocabulary related to a generic domain such as medicine, or automobiles.

OWL, the Web Ontology Language, has been defined by the Web Ontology Working Group of the W3C[2]; this language has been proposed as a standard for Web ontologies. It allows us to describe the semantics of knowledge in a machine-accessible way. OWL builds upon RDF and RDF Schema, and provides an additional vocabulary to formalize a domain by defining classes and properties of those classes (Antoniou and van Harmelen, 2004).

"Design" is a specific type of domain knowledge. This knowledge can be represented by an ontology. Architectural design is one sample. We can discover from recent research (Lin and Chiu, 2005; Hwang and Choi, 2003; Kraft and Schneider, 2005) how researchers have tried to represent design knowledge.

Adaptive Recommender System and its Techniques

Adaptive hypermedia systems build a model of the goals, preferences and knowledge of each individual user, and use this model throughout interaction with that user in order to adapt to the needs of the user (Brusilovsky, 2001). There were several kinds of adaptive hypermedia system since 1996, such as educational hypermedia, on-line information systems, online help systems, and information retrieval and filtering hypermedia. *Web-based adaptive educational hypermedia systems* provide complete frameworks and even authoring tools for developing Web-based courses, such as ELM-ART, InterBook, AHA, and KBS-Hyperbook. Adaptive information filtering is a classic technology from the field of information retrieval. Its goal is to find a few items that are relevant to the user's interests out of a large pool of documents. On the Web, this technology has been used in the context of both searching and browsing. It has been applied to adapt the results of Web searching using filtering and ordering and to recommend the most relevant documents in the pool using link generation (Brusilovsky and Peylo, 2003).

Recommender systems are a special type of information-filtering system. Recommender systems provide personalized recommendations based on the user's preferences. User profiles are important in recommender system. A user profile contains static demographic information such as name, age, country and education for each user of a Web site, as well as information about the user's interests and preferences. Such information is acquired through registration forms or questionnaires, or can be inferred by analyzing Web usage logs (Eirinaki and Vazirgiannis, 2003). We can obtain *explicit* information from user's registration data, and obtain implicit information by recording the navigational behavior or preferences of each user. We can discover users' behavior patterns by applying Web-mining techniques. There are three quite different activities in Web mining: structure mining, usage mining and content mining (Linoff and Berry, 2001).

Recommender systems collect user feedback in the form of ratings for items in a given domain and exploit similarities and differences in the profiles of users to determine how to recommend an item (Sugiyama et al., 2004). There are three common types of recommender system: content-based, collaborative filtering, and hybrid recommendation. The *content-based* approach analyzes the user's preferences and measures the similarity of various items to those preferences. Items that have a high degree of similarity to the user's preferences are recommended to the user. However, this approach has limitations: (a) we must predefine the attributes of all items, and (b) this approach cannot filter the quality and style of items. *Collaborative filtering* is an approach that utilizes explicit or implicit ratings from many users (i.e., it uses a collection of user profiles) to recommend items to a given user (Popescul et al, 2001). The items usually needs to be rated by the user on the basis of his/her likes or dislikes. However, explicit rating may have a problem of sparseness of ratings. *Hybrid*

2 www.w3.org

Table 1
An example of the categories of an architectural case.

VRA Core Categories	Possible Descriptors (sub categories)	Architectural Case Example	AAT
Record Type		work	
Type		single-family house	ID: 300122195
		terrace house	ID: 300132989
		cantilever	ID: 300007845
		house	ID: 300005433
Title	Title.Translation Title.Series	Fallingwater	
Material	Material.Medium Material.Support	concrete	ID: 300010737
		stone	ID: 300011176
Technique		cantilever construction	ID: 300135410
Creator	Creator.Role	architect	
	Creator.Personal name	Frank Lloyd Wright	
	Date.Creation Date.Restoration	1936	
Measurements, ID Number, Culture, Subject, Source, Relation, Description…			
Location	Location.Current Site	United States, Ohiopyle	
	Location.Built Environment	hillside	ID: 300008033
		slope	ID: 300132356
		rural	ID: 300229355
Style/Period	Style/Period.Style Style/Period.Period	Prairie Style	ID: 300018208
		Expressionist	ID: 300021502

recommender systems attempt to combine the content-based and collaborative filtering approaches to eliminate their disadvantages (Balabanovic and Shoham, 1997).

Methodology

Ontological Description of Architectural Cases

In order to build a Semantic Web that consists of many architectural cases, we need to annotate architectural cases. We have used VRA Core Categories, Version 3.0[3] as data elements to describe the properties of an architectural case. The categories included *Record Type, Type, Title, Measurements, Material, Technique, Creator, Date, Location, ID Number, Style/Period, Culture, Subject, Relation, Description, Source, and Rights*. Besides these 17 classes, we can add other classes or subclasses. In this study, we added several descriptors. For example, "Creator" was subdivided into Creator.Personal name, Creator.Corporate name, and Creator.Role, etc.; "Measurements" was subdivided into "Measurements.Shape", "Measurements.Dimensions", "Measurements.Format", etc. The allowed values of the descriptor "Material" included "steel", "iron", "brick", "glass", "cable", etc. Table 1 shows an example of those categories using the Fallingwater as an architectural case.

3 www.vraweb.org/vracore3.htm

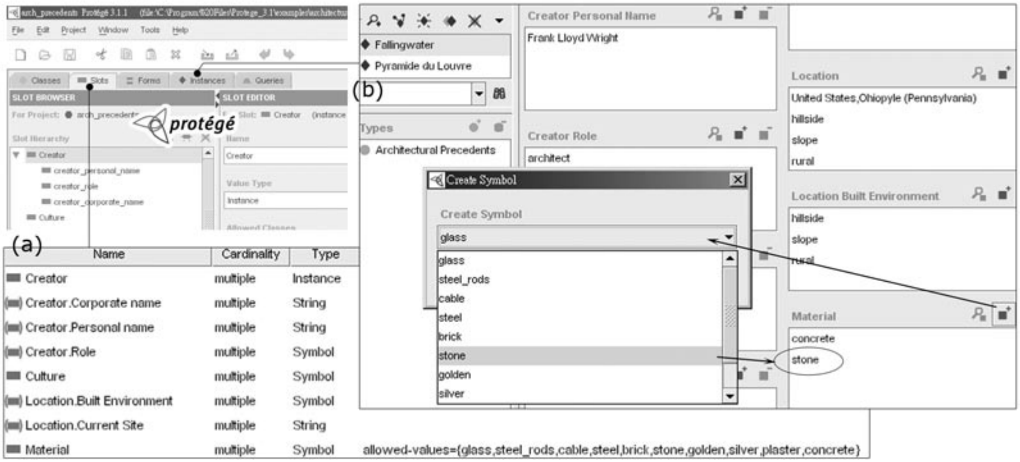

Figure 2
Protégé editor: (a) template for description of architectural cases; (b) the Fallingwater as an architectural case.

Figure 2 shows the template for the description of architectural cases that we developed for describing specific cases. In addition, we used The Art and Architecture Thesaurus (AAT) (www.getty.edu/research/tools/vocabulary/aat) to provide metadata standards. AAT is a structured vocabulary from the Getty Trust of more than 125,000 terms for describing fine art, architecture, the decorative arts, archival materials, and material culture. The data values of the VRA Core Categories are recommended for use as standards with AAT.

We used the Protégé-OWL editor (protege.stanford.edu) to build ontologies of architectural cases for the semantic web. After we had built the ontologies, we used Protégé-OWL to generate code as OWL. OWL is an ontological language that describes architectural cases. Then, we used Jena, which is a plug-in for the Protégé editor, to export OWL model in the Jena format and store it in a MySQL relational database.

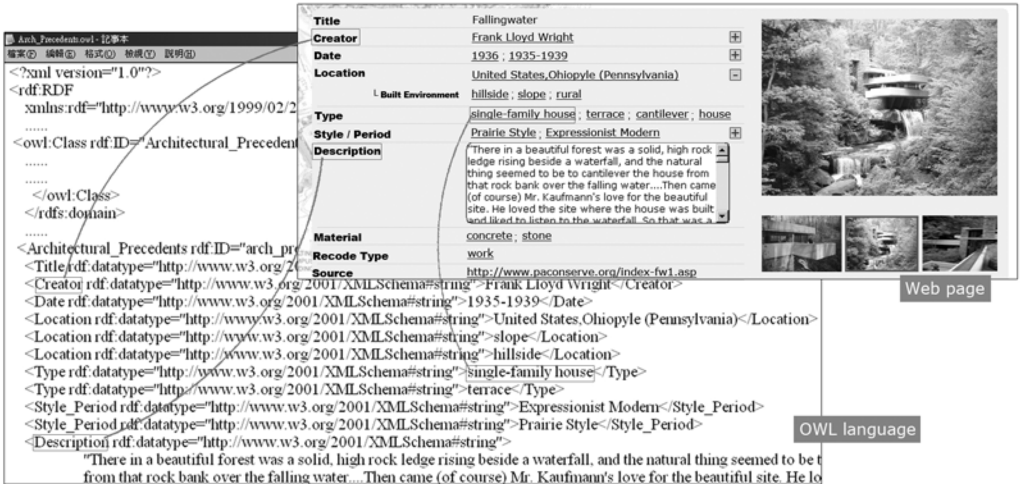

Figure 3
Part of the OWL code used to describe one architectural case – the Fallingwater.

The content of our adaptive recommender system was built on the basis of OWL. The system can show clearly the architectural domain knowledge on the Web and support users' browsing. Users also can search architectural cases by semantic keywords. Figure 3 shows part of the OWL code used to describe the structure of a Web page.

Hybrid Recommendation Approach with Semantic Query in eCAADe

We apply the *hybrid* recommendation approach, which combines the content-based and collaborative filtering approaches, in our system (Pan and Lee, 2006). The process used by the recommendation system can be explained as follows.

(1) *A user profile model* stores estimates of users' preferences for architectural design cases. The user profile model is constructed by use of explicit user feedback (e.g., users' ratings of cases) and implicit feedback (e.g., users' navigation behavior when collecting architectural cases). *Explicit feedback* can be measured by rating the degree of preference on the basis of users' likes and dislikes. We used a five-point Likert scales methodology to measure users' ratings for each case. To obtain the *implicit feedback*, we wrote a cookie program to record the data we needed and applied data-mining techniques to discover users' behavior patterns when they navigated the website. From the resulting information, we were able to determine users' viewing times and frequencies for each architectural case. If a case was viewed by users frequently and if users spent a lot of time at this site, we could assume that users had high interest in that case.

(2) Semantic queries can help a user to obtain data from a database without knowing its detailed syntactic structure. In our study, we combined keyword search with ontology domain knowledge. Users could search the architectural cases by keyword semantic queries. This method can not only search for results that match keywords, but also search for other results related to keyword. On the basis, we were able to determine users' preferences according to the frequency of browsing of ontology classes.

(3) The system measures the similarity between different users by computing the Pearson correlation coefficient r: The similarity between an active user s_x and another user s_y is calculated as follows:

$$\mathrm{sim}(s_x, s_y) = \frac{\sum_i^m (x_i - x_{avg})(y_i - y_{avg})}{\sqrt{\sum_i^m (x_i - x_{avg})^2}\sqrt{\sum_i^m (y_i - y_{avg})^2}} \quad (1)$$

Here x_i denotes the preference rating of the user s_x for architectural case i, x_{avg} is the average preference rating of the user s_x, and m is the number of architectural cases co-rated by both user s_x and user s_y.

(4) According to the result of step (3), architectural cases are recommended to the active user.

System Implementation

Figure 4 shows the architecture of our system. The eCAADe system is divided into five parts: (1) User profile module: this module acquires preferences from users, combines implicit feedback and explicit feedback and stores them; (2) Recommendation engine module: it measures the similarity between different users and recommends relevant architectural cases to users from the results; (3) Query analysis module: according to users' query, the module analyzes the query and retrieve data using OWL. It also connects the user's query and ontological knowledge and store the information in the user profile; (4) OWL convert module: we have archived 300 architectural cases to manipulate. Using Protégé editor we describe those architectural cases and convert it as OWL; (5) Ontological case base module: it stores the cases represented by the OWL. It supports users to navigate and query for architectural cases.

Figure 5 shows the main interface for the users to query and get recommendations.

Semantic query part supports users to search architectural cases allowing users to exploit the items, which are derived from class template. Recommendation part displays the recommended architectural

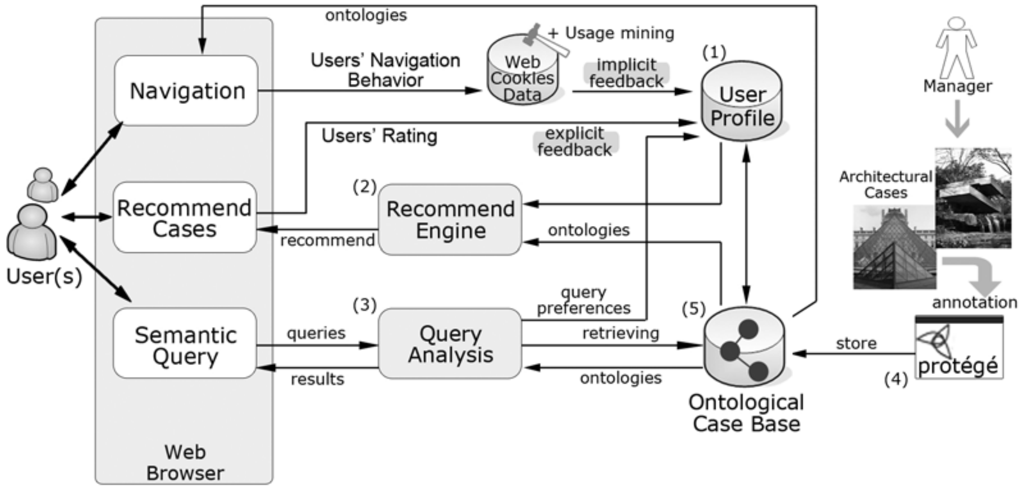

Figure 4
System architecture of the eCAADe.

cases for users by ratings. The active user can add his/her own rating on the basis of his/her preference.

Lesson Learned

We build the eCAADe system to support students to get efficient and precise recommendation during the case study phase of an architectural design process. We have used ontological concept to describe architectural design cases and implemented the design concept with semantic query on Semantic Web. This is helpful to get users' implicit feedback to make recommendation mechanism precise because the Semantic Web structure can let us know the users' navigation behavior when collecting architectural cases. We also apply the hybrid recommendation approach to combine the content-based and collaborative filtering approaches to get benefits from both sides and to reduce the innate difficulties of estimating users' original intentions.

In the future, we want to do some experiments to prove whether our recommendation system is

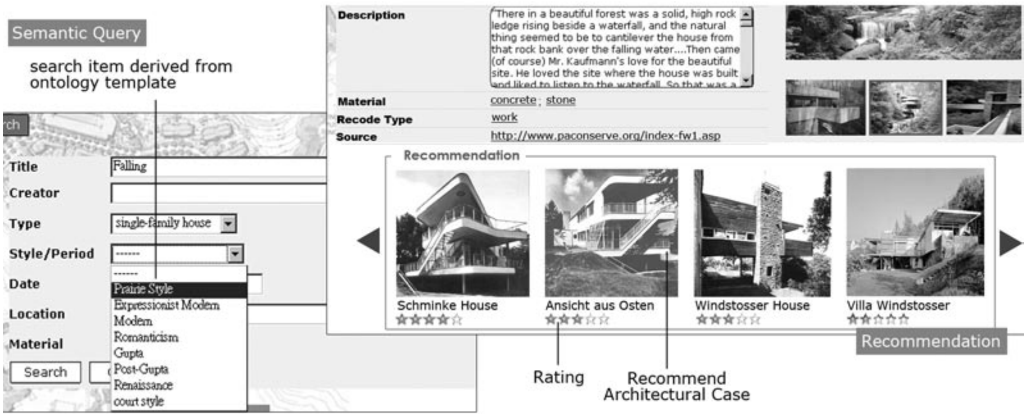

Figure 5
A snapshot of the interface – semantic query and recommendation.

more efficient and precise than others, which do not have recommendation mechanism, or not.

References

Antoniou, G. and van Harmelen, F. (ed.): 2004, A Semantic Web Primer, The MIT Press, Cambridge, MA.

Balabanovic, M. and Shoham, Y.: 1997, Fab: Content-Based, Collaborative Recommendation, Communications of the ACM, 40(3), pp. 66-72.

Brusilovsky, P. and Peylo, C.: 2003, Adaptive and Intelligent Web-Based Educational Systems, International Journal of Artificial Intelligence in Education, 13(2-4):, pp. 156-169.

Brusilovsky, P.: 2001, Adaptive hypermedia, User Modeling and User Adapted Interaction, 11(1-2): 87-110.

Eirinaki, M. and Vazirgiannis, M.: 2003, Web Mining for Web Personalization, ACM Transactions on Internet Technology, 3(1), pp. 1-27.

Fensel, D., Hendler, J., Lieberman, H. and Wahlster, W. (ed.): 2005, Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, The MIT Press, Cambridge, MA.

Gruber, T.: 1993, A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, 5(2), pp. 199-220.

Guarino, N.: 1998, Formal Ontology and Information Systems, in N. Guarino (ed.), Proceedings of the 1st International Conference in Formal Ontology in Information Systems, pp. 3-15.

Heylighen, A.: 2000. In case of architectural design. Critique and praise of Case-Based Design in architecture, PhD Dissertation, K.U.Leuven, Fac. of Engineering, Dept. of Architecture, Leuven.

Hwang, J.-E. and Choi, J.-W.: 2003, SpaceScope: Spatial Content-based Retrieval of Architectural Floor Plans, in M.-L. Chiu, J.-Y. Tsou, T. Kvan, M. Morozumi and T.-S. Jeng (eds.), Proceedings of the 10th International Conference on Computer Aided Architectural Design Futures, pp: 53-62.

Kraft, B. and Schneider, G.: 2005, Semantic Roomobjects for Conceptual Design Support: A Knowledge-based Approach, in B. Martens and A. Brown (eds.), Proceedings of the 11th International Conference on Computer Aided Architectural Design Futures, pp: 207-216.

Lin, C.-J. and Chiu, M.-L.: 2003, Smart Semantic Query of Design Information in a Case Library, in M.-L. Chiu, J.-Y. Tsou, T. Kvan, M. Morozumi and T.-S. Jeng (eds.), Proceedings of the 10th International Conference on Computer Aided Architectural Design Futures, pp: 125-136.

Lin, C.-Y. and Chiu, M.-L.: 2005, The Concept Space Explorer, in B. Martens and A. Brown (eds.), Proceedings of the 11th International Conference on Computer Aided Architectural Design Futures, pp: 139-148.

Linoff, G. and Berry, M. (ed.): 2001, Mining the Web: Transforming Customer Data, Wiley, New York.

Pan, S.-F. and Lee, J.-H.: 2006, eDAADe: An Adaptive Recommendation System for Comparison and Analysis of Architectural Precedents, in V. Wade, H. Ashman and B. Smyth (eds.), Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Springer-Verlag, Berlin, pp: 370-373.

Popescul, A., Ungar, L., Pennock, D. and Lawrence, S.: 2001, Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments, in J. S. Breese and D. Koller (eds.), Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, pp: 437-444.

Sugiyama, K., Hatano, K. and Yoshikawa, M.: 2004, Adaptive web search based on user profile constructed without any effort from users, in S. I. Feldman, M. Uretsky, M. Najork and C. E. Wills (eds.), Proceedings of the 13th international conference on World Wide Web, pp: 675-684.

Zimring, C., Bafna, S. and Do, E.Y.-L.: 1996, Structuring Cases in a Case-Based Design Aid, in J. Vanegas and P. Chinowsky (eds.), Proceedings of the third Congress on Design Computing, A/C/E'96, American Society of Civil Engineers, Anaheim, California, pp. 308-313.