Ontology-based multimedia data mining for design information retrieval

Simeon J. Simoff and Mary Lou Maher[i]

Abstract.

Design information systems - rich multimedia content with symbolic retrieval

Design information systems serve different purposes during different phases of the design process. During the early conceptual design stage the information they define the required function and the constraints, and sketches indicate the alternatives considered. Once a partial design is developed, they provide information for its further elaboration. During later stages designers need information about the physical components for the detailing the design solution. If the final design is used and constructed, there is also a need to collect and file related information for later recall. It is not surprising that design information systems probably span the whole variety of today's multimedia information systems, as shown in Figure 1.

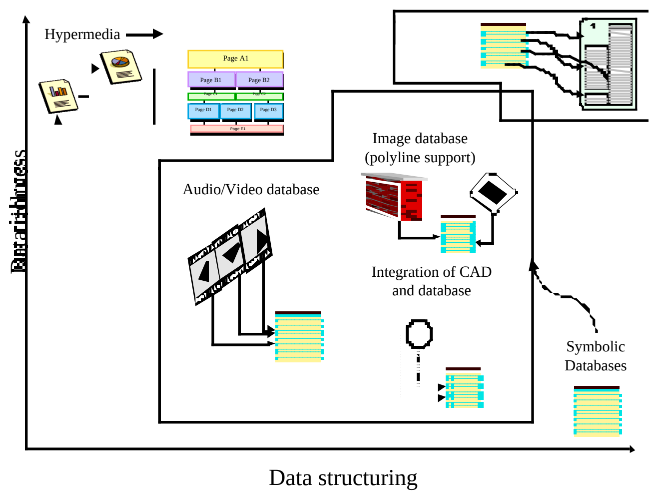

Figure 1. Design information systems

Different forms vary with respect to structuring and richness of design data. Design domain knowledge is coded in a variety of machine readable forms: as electronic manuals and references, CAD drawings and images. Thus multimedia design representations encompass a variety of data:

- structure-valued data, including: (*i*) attribute-value pairs; (*ii*) relational tables; (*iii*) object-oriented data structures;
- weakly structured data, including: (*i*) texts in free or table format; (*iii*) vector graphics, such as CAD drawings, object-oriented images;
- raw data, including: (*i*) raster images of photographs, sketches; (*ii*) animated images; (*iii*) audio and video data;

[i] Key Centre of Design Computing, University of Sydney, NSW 2006, Australia

- links data, including: (*i*) hyperlinks within weakly structured data; (*ii*) links between structure-valued components and elements of weakly structured data; (*iii*) links within structure-valued data; (*iv*) information about the sequence of visited links.

Conventional databases provide rigid structured data with well-developed retrieval mechanisms, based on predefined keys or keywords. However, classical databases have limited expressiveness.

Multimedia design information systems are frequently viewed as the backbone for design collaboration support in a multi-disciplinary project environment (Fruchter *et al.*, 1996). These systems can be viewed as databases extended with connectivity to objects in CAD systems or with ability to store images, audio and video data. In any case, retrieval is based on indexing and querying corresponding symbolic information stored in the database. For example, images, individual and group video scenes are annotated either by a set of key words or by a free text description. Part of the CAD drawings could have an underlying representation in the form of polyline coordinates. The development of the hypermedia systems where the information is organized as a set of linked pages slightly changed the picture: the keywords are matched against the text in the pages of the document and the retrieval of the page is based on different measures of its relevance to the topic (see Green and Edwards, 1996). More sophisticated models use multi-level retrieval, where at the low level algorithms are implementing different strategies between narrow, focussed and exhaustive retrieval.

Various data types have different semantics associated with them. Visual data are perceived differently by different people, thus, we end up with numerous interpretations of the same data, which is difficult to express by a flat collection of keywords. We can take advantage of the rich semantics by providing a domain-oriented multimedia data model based on the semantics associated with each data type.

Thus, the usefulness of retrieved design information strongly depends both on knowledge about the design discourse and domain, and about designer's style and background. This brings a variety of indexing challenges for context-based retrieval in design information systems. Content-based retrieval addresses the problem of integrated access to structured data and text sources as well as multimedia raw data. However, the large information content present in a multimedia data makes manual indexing labor intensive, time consuming and prone to errors.

Data mining in design information systems

A promising way to override these difficulties is to employ data mining techniques. Data mining (DM), known also as knowledge discovery (KD), is the overall process of examining a data source for implicit information and recording this information in explicit form, in other words, the extraction of a high-level knowledge from a low-level data. KD involves the identification of potentially useful and understandable patterns in this data (Fayyad *et al.*, 1996; Holland, 1986), spanning the entire spectrum from discovering information of which one has no knowledge to where one merely confirms a well known fact.

Historically, the idea to unlock the active information that is buried in the millions of data records has appeared as a response to the call for techniques and tools for data analysis which go beyond the standard query-retrieval mechanism and basic graphic capabilities for business data processing. Data mining applied methods have been developed in machine learning, statistics, data visualization and deductive databases to examine the content of large databases, i.e. for structure-valued data (Chen *et al*., 1996).

However, a vast amount of design specialised knowledge is coded in machine readable textual form or as electronic dictionaries, manuals and references, as CAD drawings or digital images. The straightforward application of data mining techniques to the variety of design data is not an easy task. How to connect the patterns, extracted from different “media”, how to represent and incorporate them into existing information structures, what kind of indexing schema to apply for retrieving relevant data?

We consider that the KD process in multimedia design data has to take in account the underlying information model. Unfortunately, there are numerous information models. The "product modeling" approach is one of the most popular streams. The aim of "product modeling" projects like STEP (Wilson, 1993), RATAS (Björk, 1994) and COMBINE (Augenbroe, 1996), was the development of integrated building information model which could serve as an integrating layer in design environment or information system. Building models could provide a common basis for communication but despite many efforts, there is still disagreement about the capacity for expression and standardisation of building representations, not only during the many stages of design and construction, but also for the different engineering contexts. The emphasis of building product models is still only on the spatial layout and building structural information (Dias, 1996). Recent works (Fenves *et. al.*, 1994), (Brandon and Betts, 1995) identify the need for developing a conceptual framework for comprehensive integrated and coordinated information systems in the Architecture/Engineering/Construction (AEC) industry to optimise the benefits of integrated information modeling (IIM) in contemporary and future technology.

The model of the *design case* is another popular and more promising way of organizing multimedia information in design information systems. The case metaphor allows incorporating in a single unit both the set of design requirements, i.e. goals, functional requirements and constraints, and the set of design solutions. The flexibility of case representation spans the possible information organization from flat attribute-value structures to multi-layered hypermedia (Maher, 1997).

The data organization units in database mining are the *data columns*. In an information model based on the design case metaphor, the organizational unit is the *case*, which defines the context of the data. In addition to identifying knowledge for improving the indexing and query formulation, knowledge discovery in design case libraries has the potential to improve the overall case representation and case content. Techniques developed for data mining are applicable to the structure-valued case data. Weakly structured and raw data require the adaptation of existing methods for text, image, video and audio fragment analyses, and the development of new specific methods, which also take into account related links between structured and non-structured data.

As illustrated in Figure 2, the information model constitutes the initial basis for the multimedia data mining. The model is a source for the terminology, the distinct attributes and the corresponding raw data, and specifying the scope of the investigation. The model acts as an initial hierarchy and the links between them.

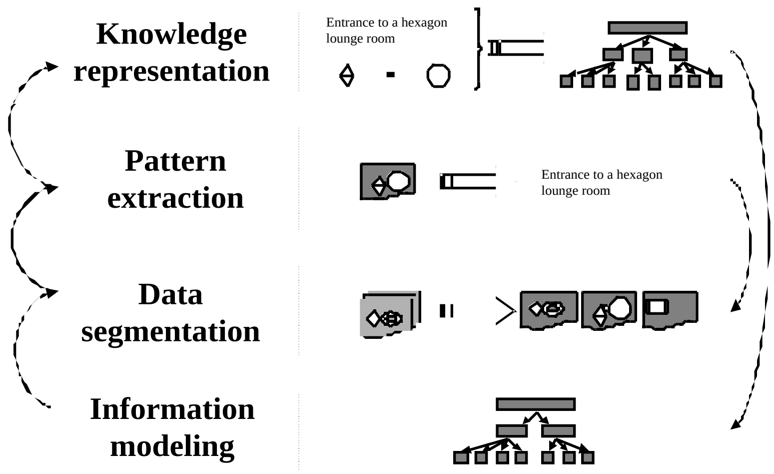

Figure 2. Multimedia data mining of design data.

During data segmentation, multimedia design data are broken into logical interconnected segments. For example, in a hypermedia case library each segment can include one or more pages. A page segmentation can group an image and the portion of the text relevant to that image. A text segment could be a paragraph, a sentence. During the pattern extraction stage the content information is retrieved and then represented as text, strings or data tokens which can be used for indexing and retrieving. Finally extracted patterns are incorporated and linked under the framework of the information model. As a result there can be additional attributes, a change in links, and some attributes, paragraphs, images or other media could become insignificant. Consequently, the information model should be able to accommodate changes in the structure and media content. Thus it seems reasonable to view knowledge acquisition in design as information modeling.

Design ontology as an integrated knowledge model

This paradigm shift has lead to the recognition of a need for an integrating knowledge model. We consider design ontology to become the integrating framework at different levels of abstraction in data modeling.

Ontology has originated in philosophy as a systematic account on the nature and the organisation of reality. The etymology of the word ontology (onto - being, logia - world, discourse) refers to the existence of the world. The concept of ontology entered the field of artificial intelligence as a formal system for representing domain concepts and their related linguistic realizations by means of basic elements. Unfortunately, there is also a growing confusion about the meaning of the term in the context of its usage in design. The notion of design ontology spans from a STEP product model to a concept structure for sharing ideas in design collaboration. In the data mining context, ontology is viewed as a formal structure or system which encapsulates the semantics of domain conceptualisation. In this sense, the ontology defines the semantics of what is known about the design domain that the ontology covers.

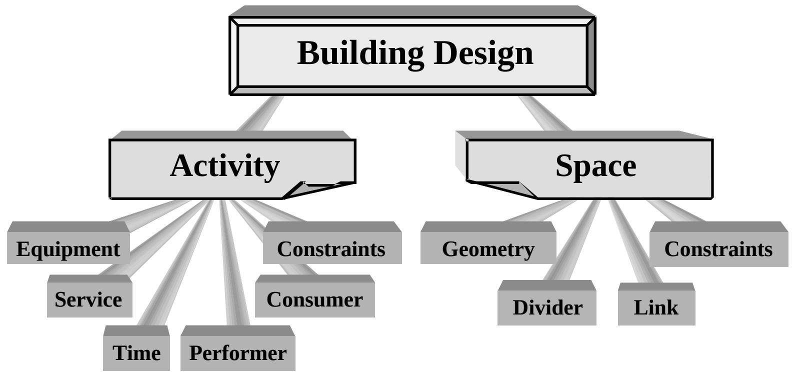

Figure 3. Architectural building design ontology

The formal, explicit representation of structure and semantics in building design is captured by the Activity/Space design ontology, presented in Figure 3. The same ontology can be expressed in symbolic form (see Maher *et al.*, 1997). A characteristic of a design ontology is the notion of change, the change in design knowledge as more experience is gained, as well as the changing model or perception of a design while designing. Our observation is that current ontological representations in the design domain are static (Simoff and Maher, 1998). The Activity/Space design ontology was proposed as a dynamic approximation of design domain knowledge. Such an approximation provides the ability both to accommodate changes in meaning during the evolution of a design idea and to maintain these changes at the levels of the data model and data structures.

Activity/Space ontology as a knowledge model defines building design in architectural terms of spaces and activities performed in these spaces. Another building design ontology, labeled Vertical/Lateral/Footings, expresses building design in terms of structural engineering systems. Its top level is shown in Figure 4. This ontology is part of the knowledge model which integrates design information in SAM design cases (Maher, 1997).

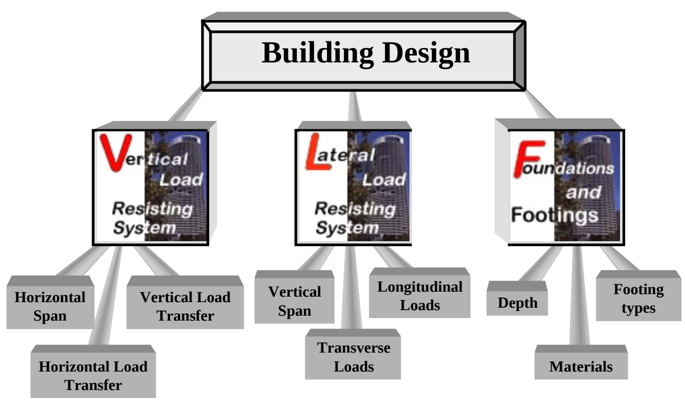

Figure 4. Structural engineering building design ontology

Algorithms for construction of ontology systems are based on the componential analysis, which, in some sense, is similar to the object-oriented analysis. The sense of a term is defined as a set of *features* that distinguish it from other terms in described domain. The whole set of values of all features automatically composes the thesaurus. Thus as a formal system, ontology includes:

1. *basis* - sets of structured and typed entities;
2. a finite set of *basic predicates* describing elementary relations, facts and actions and their related properties;
3. set of functions operating on entities, expressed through the basic predicates These functions are usually defined from text analysis.

Ontological systems are constructed as a sub-set of first order logic. They represent linguistic as well as domain-dependent knowledge at various levels of generality. The main point in constructing an ontology is to be able to define a real basis of knowledge associated to the applied domain. The basic variables then are used to label the type of an entity or an argument in a predicate. Arguments may be mono- or polymorphic.

We illustrate the concept by constructing another simplified ontology for a portion of the building design domain covered in SAM. We compose the following ontology basis:

E. experts, including 4 sub-types: architects, designers, civil and electrical engineers;

I. institutions, including 3 sub-types: building construction companies, architectural firms, universities;

O. objects, including 3 subtypes: projects, buildings, structural systems;

M. structural materials, such as reinforced concrete, steel;

S. various supports, like documentation, money transactions;

F. finances - vector space of positive rational numbers (rounded-off to two decimal digits), includes strict binary relation “equal”, order relation “less than” and standard arithmetic operations.

R. order set of real numerical values.

For instance, let us have the following predicates:

assign_to (expert: E, object: O, date: R)

use_in (project: O, structural systems: O, finances: F, company: I)

made_of (structural systems: O; structural materials: M)

complete (object: O, expert: E, institution: I, date: R)

Note that predicates **assign_to** and **complete** keep explicit polymorphism with respect to objects. Using these predicates we are able to code, for instance, the information from the text “*the engineer decided to use a rigid frame system made of reinforced concrete*” as:

$$\textbf{solution}\left(e_l, o_l, o_k, m_k\right) = \textbf{assign\_to}\left(e_l, o_l, r_l\right) \text{ô}\ \textbf{use\_in}\left(o_l, o_k, f_l, i_l\right) \text{ô}\ \textbf{made\_of}\left(o_k, m_k\right)$$

The example construction of ontological representations is a matter of investigator’s intuition and various heuristics rather than formalised procedures. In SAM, a case description amalgamates a substantial amount of descriptive knowledge in unstructured text format. For this reason case retrieval algorithms include a keyword search within case text pages. The standard retrieval in most systems is based on the *boolean model*. Queries are formulated as a set of terms connected by boolean operators, typically, “and”, “or”, “not”. The case library is searched for the existence or non-existence of terms evaluating the boolean expression stated in the query. The model works quite well for specific queries with exact match, for example, the name of the architect or the building. However, even experienced users find it difficult to formulate a set of keywords that compose an “efficient” query, resulting in an appropriate number of retrieved cases that are relevant to the problem. More sophisticated retrieval techniques deal with word score metrics, weighted terms and similarity functions. These techniques raise the issues of estimating terms relevance score, the weight of the term in different cases and different problem formulations, and the adequacy of the similarity measure.

Ontology-based data mining

Pattern extraction and knowledge representation in ontology-based data mining roughly includes:

- the construction of a thesaurus, whose components and their combinations of compose case indices;
- the approximation of semantic relations between its terms, and between its terms and relevant terms, left outside the ontology;
- the extraction of patterns from structured data, CAD drawings and images and incorporation of this patterns into case indices.

For the purpose of illustration, we limit our discussion to non-hierarchical relations in narrative text descriptions and extraction of patterns from structured data. We assume that the thesaurus has been constructed. We illustrate relevant topics with examples from the SAM building design case library (Maher, 1997).

Mining non-hierarchical relations in narrative textual data

The main aim in this case is to achieve some level of disambiguation of text description by unifying the terminology that is used. The practical need comes from the nature of distributed design information systems, where the content of the systems is updated and extended by people with different background[ii].

The lexical meaning of two words w_1 and w_2 can be in one of the following relations:

Synonymy: $w_1 \quad w_2|\alpha$. Two words are synonyms if they have a significant similar semantic content in a concrete context. Semantic similarity is estimated through some *degree of synonymy* α ë $[0;1]$, which allows us to rank synonyms.

Identity: $w_1 = w_2$. A rare case of synonymy with $\alpha = 1$, which can exist in a particular context. For example, in the context of building structure notions “support” and “buttress” are identical.

Antonymy: $w_1 \leq ? \; w_2 \mid \beta$. Two words are antonyms if they have most semantic dimensions in common but they differ in a significant way on at least one essential semantic feature. Similar to synonyms, antonyms are highly contextual. They also have various degrees of opposition β . Antonyms do not necessarily divide the conceptual space into two mutually exclusive compartments which cover the whole conceptual domain.

An relatively simple class of antonyms are the *directional opposites*. They represent either basic, topological, or conceptual (metaphorical) directional oppositions, for instance, “left/right” orientation.

Complementarity: $w_1 <> w_2$. A rare class of antonyms which divide the whole conceptual space into two non-overlapping compartments. For instance, the notion pair “wide-span”/”high rise” is used to partition the SAM case library into two separate categories.

Synonyms and antonyms in information retrieval can play the role of a kind of integrity constraints about the feature-values that may be assigned to two ontology components stated as synonyms or antonyms. In practical computer lexical analysis, synonymy is measured through substitutability (Church *et al.* 1994). The idea is to approximate semantic relationship with statistical one. A word is substitutable for another word if its substitution does not change the relations in the lexical structures in which the original word occurs. In other words, let $w_1 w_i w_3$ and $w_1 w_j w_3$ be two word combinations with similar meaning in given context. Then the less frequent word w_i , can be replaced by the more frequent word w_j . Unlike synonymy, substitutability is not symmetric, i.e. inverse substitution is

---

[ii]

incorrect. For instance, in SAM cases the notion “access” can replace the notion “entrance”, but not the inverse: “access” is used more frequently and participates in additional word combinations like “road access”. In general, the linguistic and analytic methods for defining these relations remain to be developed more precisely.

In addition to information retrieval, non-hierarchical semantic approximations are useful for refining and improving design case description and page readability. Tuning readability is based on several linguistic indices, which combine two estimates:

- *word complexity*, determined either as a function of the number of syllables per word (e.g., in Flesch and Flesch-Kincaid estimators), or characters per word (e.g. in Coleman-Liau and Bormuth estimators);
- *sentence complexity*, estimated as a function of the number of words per sentences.

Extraction of patterns from structured data

The essential assumption in attribute-oriented induction is that there exist implicit relations between case attributes, which are not taken into account by the case retrieval strategies. Some issues of attribute-oriented induction in databases are examined in (Han and Fu, 1996). We discuss the issues of extracting statistical relationships, rules and qualitative functional relationships.

Attributes describe different properties of the design entities that are represented in the case library. A relational table is an extension of the attribute-value representation. It contains a collection of related attributes. Each attribute has its own domain of feasible values, thus can be considered as a separate statistical variable. Consequently, each row in the table can be considered as an observation vector of a set of variables and the whole relational table - as a statistical population. Thus, structured case data is a source for discovery of various *statistical relations*. The way to outline these relations depends on the nature of the attributes, their representation and the type of errors in their values.

If the case data consists of numerical attributes we seek to find the dependency $y(\mathbf{x})$, where y is the dependent (objective) attribute, and $\mathbf{x} = x_1, x_2, ..., x_m$ is the vector of explanatory (descriptive) attributes. Depending on the nature of the errors, we can employ either classical, non-parametric, interval or fuzzy regression analysis. Derived relationships can be used

(i) to estimate unknown or missing values in the case base, for instance, a time-series dependency between project year and the average number of stories in high rise buildings can give an estimate for a missing number of stories;

(ii) to detect the occurrence of errors in particular case records.

Note, that in the second case, there could be a situation when the values of all attributes fall within their respective domain of definition and still their combination violates derived statistical constraint.

In the case when attribute y is a categorical one, we employ grouping methods from multivariate analysis, which divide the values of vector $\mathbf{x}$ into distinct groups with respect to the corresponding dependent attribute value. These methods differ in the way they estimate similarity coefficients of a table row with respect to each individual group. Based on the similarity coefficients, we can estimate the probability that a particular row belongs to a group. Similar methods are used for the case when explanatory attributes are categorical.

In design information systems it is suitable sometimes to formulate knowledge in the form of *rules* because they are easy to incorporate in the retrieval module. Several kinds of rules can be discovered when analysing structured design data. Below are some of the rules that can be found in a design case library.

Association rule. This type of rule is the most popular in the data mining community, though when it comes to a formal definition, different authors give a different meaning to this label (for example, compare Mannila, 1997 and Han and Fu, 1996). Therefore, we give a formal definition of an association rule in terms of case attribute-value pairs. Following our notation, let's consider the attribute schema $S = \{\mathbf{x}, \mathbf{y}\}$, where $\mathbf{x} = x_1, x_2, ..., x_m$ and $\mathbf{y} = y_1, y_2, ..., y_n$, with *k*-records and a *pattern*, also called *relation* (Mannila, 1997), $\rho(\mathbf{x}) = \{x_i^{(p)} = c_i^{(p)}; ...; x_j^{(p)} = c_j^{(p)}\}$, where c denotes a particular value of an attribute, i.e. $\rho(\mathbf{x})$ is a concrete combination of attribute values. Let $\rho(\mathbf{x})$ occur *r*-times in the case base and let for some s times of these r there also occurs a pattern $\rho(\mathbf{y}) = \{y_i^{(p)} = d_i^{(p)}; ...; y_j^{(p)} = d_j^{(p)}\}$. Formally, an association rule is formulated as $\{\rho(\mathbf{x})\, ?\ \ \rho(\mathbf{y}), \sigma, \chi\}$, where $\sigma = \frac{r}{k}$ is called *support* and $\chi = \frac{s}{r}$ is called *confidence*, i.e. if some of attributes $\mathbf{x}$ satisfy relation $\rho(\mathbf{x})$ then some of attributes $\mathbf{y}$ tend to satisfy relation $\rho(\mathbf{y})$. Association is defined as *strong* if it has *large* support and *high* confidence. In practice, these notions indicate particular threshold values, subject to various domain heuristics. For example, an association rule can be derived between the site location and the footings of the buildings at the same site, making explicit the knowledge about the ground properties of the site.

Classification rule. This type of rule is used to classify cases based on *one determining attribute* x_i. For instance, a set of building cases can be classified with respect to their function.

Cluster description rule. This knowledge is derived in a way similar to the above discussed grouping methods, defining clusters by minimising the distance between attribute values within a cluster and the similarity between clusters. For example, a rule for defining a cluster of office buildings based on the values of several attributes (e.g, plan shape, floor area, floor system attributes). Note that it is possible that a case, whose attribute "Function" has value "office" may remain outside the cluster of office buildings. Note also the difference with the classification rule, where a case is classified as an office building if attribute "Function" is selected as determining and its value is "office".

Case trend rule. Such rules show the direction of change of some case attributes. For example, the rule that describes the major factors that influence the change of materials, used in described building cases.

Another information of practical interest is the qualitative relations between some attributes describing also the relations between ontology entities. Mining for implicit functional relations is based on the qualitative mathematical analysis (Forbus, 1984) and its application for generalising quantitative results (Hochka and Klosgen, 1991). Thus, the comparison of two attributes $x_i = \left(x_i^{(1)}, x_i^{(2)} \ldots, x_i^{(k)}\right)$ and $x_j = \left(x_j^{(1)}, x_j^{(2)} \ldots, x_j^{(k)}\right)$ starts with dividing their domain into distinct qualitative regions, introducing the corresponding *landmark values* as bounds of each region. Further, we compare the overall behavior of the attributes within each region. Then we derive relations between their behaviors. For example, if x_j increases then a relation of *monotonicity between attributes* x_i and x_j in a broad sense means that x_i either only increases or decreases as x_j increases (decreases). There is still no universal technique for identifying qualitative functional relationships. Existing algorithms are based on various heuristics (for example, see Zhong and Ohsuga, 1996). For instance, a qualitative relation between the attribute “height of building” and attribute “distance from the city center” can describe qualitatively the skyline of the city, assuming that attribute “city” has a constant value.

Conclusions

For the analysis of multimedia design descriptions, ontology provides a conceptual framework for a structured representation of the meaning, based on the construction of a building design thesaurus and approximation of semantic relations between its terms, and between its terms and relevant terms, left outside the thesaurus. Establishing non-hierarchical relations between terms in the ontology, like synonymy, identity, complementarity and substitutability, provides basis for improving the indexing and reusability of design documentation.

For multimedia design data mining, ontology provides context, structure and relationships for representation and integration of discovered patterns. For instance, discovering the knowledge in a floor plan document, the size, colour, number of rooms, the shape of the rooms can be related through the means of the Activity/Space ontology to their possible functionality. In some sense ontology provides a hypothesis for understanding what was done but not well documented.

The use of multimedia data mining for improving the indexing and retrieval in design information systems is still in its infancy. The ontology-based approach has the potential for enhancing the content-based retrieving strategies adding new ways for estimating similarity between query and the design cases.

References

Augenbroe, G. (1996). COMBINE project: The broad perspective, Proceedings of the International Construction Information Technology Conference InCIT 96, Sydney, 103-108.

Björk, B.C. (1994). RATAS Project - Developing an Infrastructure for Computer Integrated Construction, Computing in Civil Engineering, **8**(4), 401-435.

Brandon, P. and Betts, M.(eds): 1995, Integrated Construction Information, E & FN Spon, an Imprint of Chapman & Hall, London.

Chen, M-S., Han, J. and Yu, P.S. (1996). Data mining: an overview from a database perspective. Knowledge and Data Engineering, **8** (6), 866-883.

Church, K.W., Gale, W., Hanks, P., Hindle, D. and Moon, R. (1994). Lexical substitutability. In B.T.S. Atkins and A. Zampolli (eds), Computational Approaches to the Lexicon, Oxford University Press, pp. 153-177.

Dias, W.P.S.: 1996, Multidisciplinary product modeling of buildings. Computing in Civil Engineering, **10**(1), 78-86.

Fenves, S., Flemming, U., Hemdrickson, C., Maher, M.L., Quadrel, R., Terk, M. and Woodbury, R.: 1994, Concurrent Computer-Integrated Building Design, Prentice Hall, Englewood Cliffs, New Jersey.

Fruchter, R., Reiner, K., Leifer, L. and Toye, G. (1996). VisionManager: A computer environment for design evolution capture. In J. S. Gero and F. Sudweeks (eds), Artificial Intelligence in Design'96, Kluwer Academic Publishers, Dordrecht, pp. 505-524.

Green, C. L. and Edwards, P. (1996). Using machine learning to enhance software tools for Internet information management. Technical Report WS-96-06. AAAI Press, Menlo Park, CA, pp. 48-55.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. (1996). From data mining to knowledge discovery: An overview. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (eds), Advances in Knowledge Discovery and Data Mining, AAAI Press, Boston, MA, pp. 1-34.

Han, J. and Fu, Y. (1996). Attribute-oriented induction in data mining. In U.M., Fayyad, G., Piatetsky-Shapiro, P., Smyth, and R. Uthurusamy (eds), Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, pp.399-424.

Hochka, P. and Klosgen, W. (1991). A support system for interpreting statistical data. In G. Piatetsky-Shapiro and W. J. Frawley (eds), Knowledge Discovery in Databases, AAAI Press/ The MIT Press, Cambridge, Massachusetts, pp. 325-345.

Holland, J.H., Holyoak, K.J., Nisbett, R.E. and Thagard, P.R. (1986). Induction: Processes of inference, learning and discovery, Computational Models of Cognition and Perception, The MIT Press, Cambridge,.MA.

Maher, M.L. (1997). SAM: A multimedia case library of structural designs. In Y.T.Liu, J-Y. Tsou and J-H. Hou (eds), CAADRIA'97, Hu's Publisher Inc., Taiwan, pp. 5-14.

Maher, M.L. and Simoff, S. (1997). Knowledge discovery in multimedia design case bases. In B. Verma and X. Yao (eds), Proceedings ICCIMA'97, Griffith University, Gold Coast, pp. 6-11.

Maher, M. L., Simoff, S. J. and Mitchell, J. (1997). Formalising building requirements using an Activity/Space Model. Automation in Construction, **6**, 77-95.

Mannila, H. (1997). Methods and problems in data mining. In F. Afrati and P. Kolaitis (eds), Database Theory - ICDT'97, Springer Verlag, Heildelberg, pp. 41-55.

Simoff, S. J. and Maher, M. L. (1998). Designing with the Activity/Space ontology. In J. S. Gero and F. Sudweeks (eds), Artificial Intelligence in Design'98. Kluwer Academic Publishers, Dordrecht (to appear).

Wilson, P. R. (1993). A view of STEP, in Wilson, P.R., Wozny, M.J. and Pratt, M.J. (eds), Geometric and Product Modeling, North Holland, Amsterdam.

Zhong, N. and Ohsuga, S. (1996). A multi-step process for discovering, managing and refining strong functional relations hidden in databases. In Z. W. Ras and M. Michalewicz (eds), Foundations of Intelligent Systems, Springer Verlag, Heidelberg, pp. 501-510.