How to support city planning using map interpretation techniques

Barbara Schmidt-Belz, Angi Voß, Lars Emkes, Carl-Helmut Coulon
GMD - German National Research Center for Information Technology
D-53754 Sankt Augustin

contact: Barbara Schmidt-Belz
phone: +49 - 2241 - 14 - 2796
email: Barbara.Schmidt-Belz@gmd.de
fax: +49 - 2241 - 14 - 2384

ABSTRACT:

We suggest and motivate a system to support city traffic planning. Our approach is derived from Case-Based Reasoning (CBR), where former experiences (cases) are stored and made available for reuse. To start with, a collection of examples from books or other sources is stored as hypermedia documents. Retrieval of useful examples is enabled by describing (indexing) the examples in several aspects. While some descriptors have to be attached by users or system administrators, others could be automatically inferred. The vision is, that in the long run cases are derived from GIS plans and the CBR support is an integrated tool in a GIS working environment.

1 MOTIVATION

As conditions and policies on traffic and urban mobility change, city planners are confronted with ever new requirements and problems. Though they are experts in the traditional way of planning in their town, they are faced with new problems they are not experts in. But there may exist expertise elsewhere, in another department, town or country. This paper explores how existing expertise could be made available in a form that is suitable for immediate reuse.

Planners traditionally are guided by examples, together with rules and constraints (legal, engineering or others). This is the knowledge they rely on. Examples are
frequently taken from personal experience, sometimes from a local fund of experience, like locally stored plans or discussions with colleagues. For new types of problems or policies experience is scarce and hardly available. It takes some years until books or articles are available, and it takes some more years until solutions are broadly perceived and used by planners. In this paper, we suggest how concrete examples that might be relevant for the problem at hand can be retrieved in a very flexible way.

In order to compare examples and to judge their relevance for the situation at hand we suggest to consider them under different aspects, for instance under the issues they tackle, the concepts they apply, the patterns or schemes they follow. Each aspect has its own vocabulary. Though initially unrelated, associative connections between these vocabularies begin to emerge as examples are being collected. These statistical associations generate a semantic net that allows to draw inferences beyond the combinations that actually occurred in the examples.

Describing an example under different aspects requires a semantic interpretation, e.g. an explanation or rationalizing, which must be left to an expert. But if plans and schemes are represented in some GIS or CAD format, they can be interpreted automatically to some extent. We developed such methods in the FABEL project for plans of complex buildings. The purpose of FABEL was to support architects and civil engineers in the retrieval and reuse of previous designs by case-based and knowledge-based methods (Voß et al. 1994).

2 A SCENARIO: CYCLE TRACK PLANNING

Let us motivate our ideas with a concrete example. As an city planner, suppose you are concerned with traffic planning. At the moment you have to solve a problem about a junction between a cycle track and a street, as sketched in figure 1. Often, cars are parked at the junction, because there is enough space left without obstructing the cars on the street.
2.1 Books

In short terms, there is a problem of "junction blocked by illegal parking". Maybe, your politicians have recently decided to make your town more attractive for bikers. So there is not yet much experience with building cycle tracks in your department. Nowadays, we could offer you books and articles like the very good Dutch book on bicycle traffic planning (C.R.O.W. 1994) - the Dutch being protagonists in this field. The book contains a lot of pictures with concrete examples, a set of more abstract schemes, constraints, and guidelines.

As a first step towards automated support, we could collect examples from books, organize them as a hypermedia document and let you browse through it. But is browsing through hypermedia so much better than browsing through a book? We think it would be, if the document grew from use. If new examples were added by other planners like you, who used the book, found useful schemes and examples, applied them to their current problem and fed their experience back into the document. The set of examples would grow, and so would the set of schemes, though at a slower pace. Moreover, examples and schemes would get stuffed with comments of how well or how bad they worked. Of course, as the repository on bicycle track planning grows
you would need better retrieval tools. Exploring the repository by hyperlink surfing would cost too much time and be of unpredictable success.

2.3 Organisation by Issues - Concepts - Examples

How else could you browse through the repository? One way would be to look up an index of issues and hopefully find the "illegal parking blocks junction" issue or, by help of a thesaurus, a related one. You could retrieve examples addressing this issue and find the first one shown in figure 2. It applies the concept "reduce space by poles" and realizes it through a scheme that inserts (or moves) some poles.

Moreover, you should notice that the same scheme realizes another concept, namely to "reduce space by narrowing" in order to address another issue, which is to "reduce velocity". For you might recognize that this issue is indeed relevant for your particular junction: you casually got complaints that bikers enter the street too fast without first looking for cars (this issue has so far been superimposed by the parking car problem). So the particular scheme you found would solve two problems at once. Looking for other solutions to the issue of "reduce velocity" you might find the second example in figure 2. But because it does not solve the car parking issue, you prefer the first one.

What we have just seen was an exploration from issues to examples and back to other concepts, issues, and examples. As an extension, given a catalogue of objects describing junctions and their components, you might describe your particular junction in this vocabulary: "a T-junction between a street and a cycle track". Then you might retrieve situations with similar objects and find out what concepts and issues are related to them. If you already have a solution in mind, you could characterize it (for example as "a T-junction between a street and a cycle track, and some poles") and retrieve similar solutions, rather than looking for similar problems. In fact, you should be able to specify any kind of information and get supplementary ones.
Figure 2: Two examples according to (C.R.O.W. 1994) with schemes and concepts that are relevant to our problem in figure 1.

"Case 1"

"before"

in spite of the pole, the junction between bike track and street invites illegal parking

"after"

this arrangement of poles would not invite illegal parking

"Case 2"

"before"

broad junction between bike track and street, bikes enter street at high speed

"after"

junction between bike track and street was narrowed down

2.4 Retrieving and reusing plans

If you are already working with a GIS, you might just indicate your current junction in some plan or draw the envisaged solution. The object types in these plan fragments could be extracted automatically, even certain important relations or complete structures. Thus, complex objects or patterns could be recognized automatically. Such descriptions would allow a more precise retrieval of similar plan fragments than, say, a concept or an issue alone.

Having explored the problem in any fashion, you will end up with some relevant issues, concepts, plan fragments or examples. If the plan fragments are in a suitable
GIS or CAD format, you could copy and paste parts from the fragments into your current plan. If pasting involves some boring and tedious adjustment of details, you might even like the selected plan fragments to be automatically pasted into your current plan.

Thus, the solution of planning problems could be simplified significantly, and playing with alternatives would be easier.

3 TOWARDS AUTOMATED SUPPORT

How realistic is such a scenario? To answer this question we derive a set of requirements in order to support such planning processes and study the more demanding ones in some detail.

3.1 Requirements

(1) A repository of examples. Ideally, examples are multimedia documents including texts, pictures, and plan fragments.

(2) A collection of examples to start with.

(3) Categories and vocabularies to describe examples. In our scenario, issues, concepts, and sets of objects were used as categories.

(4) Description of examples in terms of these vocabularies.

(5) Tools to retrieve examples using these vocabularies.

(6) A collection of plan fragments sketching or detailing problems and solutions.

(7) Conversion of plan fragments into an electronic format (in a GIS, CAD or some other object-oriented graphic format).

(8) Tools to automatically describe plan fragments.
(9) Tools to retrieve similar plan fragments.

(10) Tools to associate between descriptors.

We want to focus on retrieval: What are suitable categories to describe the examples and what vocabulary to use for each category (3)? How to find examples given partial descriptions (5)? How to describe plan fragments automatically (8) and how to retrieve them by examples (9)? How to get notified of related descriptions (10)? To approach these questions, we consider techniques from Artificial Intelligence, especially from Case-Based Reasoning as highly relevant.

3.2 Case-based reasoning

In Case-Based Reasoning, problems are solved by retrieving solutions for similar problems and adapting them to the new context (Kolodner 1993). Retrieval methods differ in the representation of the cases, the way how similarity of cases is expressed in terms of this representation, and how it is used during retrieval. So far, Case-Based Reasoning has been most successful in help-desk applications and for decision support, where no adaptation is required (Watson and Maho 1994). Adaptation methods are often domain-specific, ranging from heuristic rules to powerful problem solvers that extend, adjust or replay a case. More generic are structure mapping and transferring methods originating from the field of analogical reasoning (Gentner and Dedre 1983).

Nowadays, there is active research on applying Case-Based Reasoning to synthesis tasks, especially in design (Pu and Maho 1996). There Case-Based Reasoning is considered a key technology, because often design knowledge is only implicitly available in previous solutions, so-called precedents. But as precedents tend to be complex - like complete building models - they differ considerably from cases, which provide solutions to very specific problems. The design of a single artifact usually contains solutions to many such problems. Moreover, they usually do not have a unique representation, or such a representation is too low-level to define similarity. Similarity is a pragmatic concept, used to approximate utility (Smyth and Keane 1994). But utility depends on the task and context. To put it in one sentence: the gap between precedents and cases is often huge.
3.2.1 Categories and vocabularies to describe examples and plans (3)

Inspired by Schank (Schank 1990), Rivka Oxman suggested to describe a single precedent, which is a complex building in her domain, by a set of so-called stories. These stories should contain three sections: an issue or problem, a concept for its solution, and a form or scheme that realizes the concept and is instantiated in some part of the precedent (Oxman 1994). Stories express the design knowledge behind a precedent in chunks of convenient size, which can be treated as cases. The three categories describe a story at three different levels of abstraction, rather than directly jumping from a problem to a solution. Thus, the design process is taken into account. Finally, the three categories are quite general, so that examples from different domains might be retrieved.

We tried to apply the categories of issues, concepts, and forms to the scenario in the previous section. But probably they have to be modified. First of all, we should distinguish two situations, the original, current or problem situation and the final, envisaged or solution situation. Second, forms as copied from a book are not suitable to describe situations, because they cannot be interpreted automatically. Therefore, we should replace them by other categories such as the objects contained in the situation, or the relations it contains, or possibly the structure. The form itself would then be part of the example, which is a collection of uninterpreted multimedia documents. Third, the three categories alone might not suffice. For instance, the book on bicycle traffic (C.R.O.W. 1994) describes the examples in terms of function, usage, constraints, realization, pro and contra arguments, objects, and schemes. Function is comparable to issue, and realization to concept. The vocabulary used for each category should be broad, ranging from generic to domain-specific items. Thus, both a high recall with cross-domain references and a good precision can be achieved.

3.2.2 Tools for retrieval of examples using this vocabulary (5)

A story telling a lesson about a precedent in terms of multiple categories can be represented as a record and be considered a case. This is illustrated in Figure 3. Case-based reasoning supports queries by examples. A query is like an incomplete case. For those components of a case that can be specified in the query, concepts of similarity must be defined. As demonstrated in our scenario, design is often exploratory. There is no straight deduction from a problem to its solution. Therefore, every part of a case,
except for the example, should be admissible to contribute to a query. The planner could supply the issue he is interested in, or the concept, or the objects describing the current or envisaged situation. Therefore, similarity must be defined for each of these components. Then all components specified in a query can be compared with the corresponding components of the cases and be integrated into an overall estimate of similarity (in order to approximate relevancy or usefulness), see (Tversky 1982). Cases that are sufficiently similar are retrieved. The components which are missing in the query constitute the answer; among them is always (a reference to) the example. For instance, for a query with issues and concepts, the objects, schemes, and examples in the retrieved cases constitute an answer; for a query with objects and schemes, the issues, concepts, and examples in the cases constitute the answer.

For each of the categories issues, concepts, and object types a catalogue (or index) should be defined. Similarity could simply be defined extensionally, by filling in a (symmetric) matrix, rather than through implementing a thesaurus. A query can be specified by selecting multiple entries from the catalogues of issues and concepts, and from two instances of the catalogue of object types, one for the current situation and one for the envisaged one. Sets of objects can be specified by the number of objects of certain types. Sets of objects are considered similar if they contain similar types of objects in similar quantities. The tools needed for retrieval must compare sets of keywords and sets of quantified keywords. This is standard in information retrieval (Salton and McGill 1983) and Case-Based Reasoning (Kolodner 1993), and we used such methods in the FABEL project to retrieve similar plans of buildings (Gräther 1994) in (Voß 1994).
3.2.3 Tools for automatically describing plan fragments (8)

So far we have assumed that schemes are copied from a book and therefore are not machine-interpretable. However, if they were plan fragments in some component-oriented CAD or GIS format, they could automatically be interpreted to some extent. The same is true of plans describing the example not as a scheme, but in full detail. Given any such plan fragments, we could compare them to plan fragments specified in the query.

First of all, we could automatically extract the objects from the plans, which would avoid some of the manual descriptions assumed in (3). But we could go further and extract meaningful relations or even the main structure of the plan and store it as a graph. What relations to extract depends on the purpose of retrieval. Some relations may be directly inferable from the CAD or GIS format, others may be more complicated to compute (Theodiridis and Papadias 1995). But one should try to exploit available GIS or CAD functionality. If the object catalogue contains complex objects like "T-junction between street and cycle track" and they are not explicit in the format, one could try to recognize them from constellations of more primitive objects in the plan. In general, a trade-off between semantic contents and computation time will be encountered. In the FABEL project, we developed methods to describe and compare component-oriented CAD-plans of complex buildings as object density maps of different grain sizes (Coulon and Steffens 1994), as sets of high level gestalten (Schaaf 1994), and as structure-representing graphs (Coulon 1995). Gestalten are meaningful constellations of more primitive objects; a T-junction or an "axis of view" could be considered a gestalt.

3.2.4 Tools for retrieval by plan fragments (9)

Structures can sometimes be represented as terms, but more generally as graphs. Graph matching is NP complete. Therefore, it must be preceded by a fast filtering step. This can be done on the basis of the objects and/or relations involved, or by first comparing other parts of the query. A complete structural comparison yields the (or a) largest common substructure. This information can be used for a very precise retrieval, but is even more useful for automating the pasting of the solution to the plan at hand. We developed methods for transferring pieces of CAD plans of buildings based on structural representations, but do not consider them as relevant for immediate
commercialisation in the context of GIS. Therefore, we do not go into more details. The reader is referred to (Coulon 1995).

3.2.5 Recapitulation
So far, we have established categories of issues, concepts, (types of) objects, and, if available, of plan fragments. We have said how to define similarity between issues, between concepts, sets of objects, and plan fragments. An example is a multimedia document described by one or more cases. A case contains descriptions in several categories, for instance an issue, a concept, one or two sets of objects, possibly two plan fragments, and references to concrete examples. A query contains a selection of issues, concepts, of one or two sets of objects, possibly one or two plan fragments. The query can be compared with an example by computing their similarity in each aspect (or category) and integrating them into an overall similarity. Thus, the most similar examples can be retrieved. Such a similarity-based retrieval provides for graceful degradation, it returns cases even if none of them matches the query exactly, and even if the query contains contradictory items (which do not occur simultaneously in an example). Also the degree of similarity is returned, which provides a qualified answer.

3.2.6 Tools for association between descriptors (10)
But we can do more than that. Since each case combines items of all aspects, we can count how often a particular issue is realized by a particular concept, how often a concept is applied to or yields a particular set of objects, or how often it is realized in a particular plan fragment, etc. That means we can abstract (compute, infer, learn) associative relations between items in different categories from the cases. Together with the similarity functions defined for each category, this yields a semantic net. Its nodes are instances of the different aspects, and its links are weighted relations expressing similarity or association. By spreading activations from the items in the query through the net, we obtain similar and associated items.

This information is useful for planners, because it guides their explorations during retrieval. They are notified of items somehow related to the ones they selected and can formulate new queries to get other examples. For instance, by this means the planners in our scenario in section 2 could learn that the example they had first retrieved realized a second concept "reduce space by narrowing" addressing the issue "reduce velocity", 

238
which led them to the second example in figure 2. Furthermore, since the semantic net generalizes from the examples, it can suggest combinations of items (issues, concepts, objects, schemes) that do not exist in this combination in any example. Of course it is up to the planners to recognize such potential combinations as invalid or innovative. There are several approaches for semantic nets, for instance (Rissland et al 1995) or (Wolverton and Haeyes-Roth 1994) but we would need a very special one.

Figure 4: How retrieval and reuse of previous planning examples can be supported.
4 A PROTOTYPE IMPLEMENTATION

4.1 Putting it together

Figure 4 summarizes the ideas put forward in the previous section. Symmetric matrices define the similarity of items in each catalogue. For types of objects, quantities can be specified. The semantic net has similarity links and associative links labeled with the frequency in which the connected items are associated in the examples. Items selected in the query are highlighted and so are closely related items in the semantic network, while unrelated ones are dimmed.

A prototype demonstrating our ideas about requirements (1) - (5) and (10) is planned to be operational by July 1996. The representation, interpretation, and retrieval of plan fragments (6) - (9) will be delayed until we get access to a sufficient set of plans. The work is split into three tracks: domain-specific knowledge and data acquisition for (1) - (4) proceeds in parallel with the implementation of a generic tool for similarity-based retrieval of examples (5), which will be extented to a semantic network (10).

4.2 Domain-specific knowledge and data acquisition (1) - (4)

This track concerns the examples and the extraction of knowledge from the book on bicycle traffic planning (C.R.O.W. 1994):

• Examples:
  A supplier must be found, and the examples must be made machine-accessible.

• Knowledge:
  A set of categories (like issues, concepts, objects or others) must be decided upon. For each category, the vocabulary must be determined and similarity functions be defined. Proto-cases must be extracted from the book, be described in terms of these categories and be linked to the examples (thus getting complete cases). As a first result, figure 5 shows some cases extracted for the cycle track domain.
### Figure 5: Some cases for the cycle track planning domain.

<table>
<thead>
<tr>
<th>Issues</th>
<th>concepts</th>
<th>Objects problem</th>
<th>solution</th>
<th>schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>crossing track</td>
<td>crossing via traffic island</td>
<td>traffic island</td>
<td>127-4.52</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>speed bumps</td>
<td>separation area</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cycle track</td>
<td>road</td>
<td></td>
</tr>
<tr>
<td>crossing track</td>
<td>narrowing with marked crossing</td>
<td>sidewalk</td>
<td>projection</td>
<td>127-4.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>separation area</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cycle track</td>
<td>road</td>
<td></td>
</tr>
<tr>
<td>velocity reduction</td>
<td>narrowing with marked crossing</td>
<td>sidewalk</td>
<td>projection</td>
<td>127-4.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>separation area</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>cycle track</td>
<td>road</td>
<td></td>
</tr>
<tr>
<td>velocity reduction</td>
<td>narrowing by poles</td>
<td>cycle track</td>
<td>road</td>
<td>230-8.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pole</td>
<td>pole</td>
<td></td>
</tr>
<tr>
<td>velocity reduction</td>
<td>narrowing of junction</td>
<td>cycle track</td>
<td>sidewalk projection</td>
<td>232-8.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>road</td>
<td>cycle track</td>
<td></td>
</tr>
<tr>
<td>inhibit parking</td>
<td>exposed position of poles at</td>
<td>cycle track</td>
<td>cycle track</td>
<td>230-8.7</td>
</tr>
<tr>
<td></td>
<td>junction</td>
<td>road</td>
<td>pole</td>
<td></td>
</tr>
<tr>
<td>inhibit parking</td>
<td>narrowing of junction</td>
<td>sidewalk</td>
<td>projection</td>
<td>236-8.15</td>
</tr>
</tbody>
</table>

### 4.3 A generic tool for similarity-based retrieval of examples (5)

The tool will provide data structures for catalogues, similarities, cases, examples, and queries, as well as functions for retrieval.
Catalogues and similarities:
To define a catalogue, a set of items must be supplied and a (symmetric) matrix of similarities be filled in. It must be indicated whether items can be selected or quantified. Due to experience from the FABEL project, we will probably offer both, precise quantities and qualitative ones like few, some, many. Probably we will also offer exact and subset comparisons.

Cases and examples:
As shown in figure 3, cases are like records, their components can be typed according to a catalogue, there is a reference to external files (the examples), and there may be other components (e.g. for plan fragments) for future use.

Queries and retrieval:
For a query, the user is offered a table of catalogues, one for each comparable component of a case. So catalogues may appear more than once, for instance for the current and the envisaged situation. The user can select (and quantify) and deselect one or more entries in any of the catalogues. Selected items are highlighted.
There is another table presenting the cases retrieved in descending similarity. A case can be selected and its associated multimedia documents be inspected in a separate window.

4.4 Extension to a semantic network (10)

For the user, the extension means that upon selection of any items in the table of catalogues other, related items get highlighted as well, and unrelated ones get dimmed. The intensity of highlighting is proportional to the degree of relatedness. (Alternatively, the catalogues could be rearranged with the more highlighted items at the top. But maybe catalogues should not be rearranged dynamically.)

The semantic net contains a node for each item from a catalogue that occurred in some case. Items from the same catalogue but in different components of a case must be distinguished. Also, items in different quantities must be distinguished. Two nodes coming from the same component of any cases are related by their similarity as defined for their catalogue. Nodes from different components of the same case are related by an associative link. The link is labeled by the relative frequency of cases containing the
Figure 6: The semantic net extracted from the cases in figure 5.

pair of nodes. If a new case is inserted, the associative links for its component items must be updated in the semantic net. Figure 6 shows the semantic net (without weights) that can be extracted from the cases in figure 5.

The semantic net is assumed to have a constant energy distributed among its nodes. If an item is selected for the query, the energy of the net is decreased by some constant. This is done by proportionally lowering the energy of all nodes. The reduced amount is then added to the node selected and from there is spread to all directly linked nodes. In a second step, nodes that were activated via a similarity link pass some energy via associative links, and vice versa. This two step propagation of energy should be sufficient to raise the activation of related nodes.

5 CONCLUSION

We suggested a generic system to support city traffic planning by Case-Based Reasoning and illustrated it for the special domain of cycle track planning. Examples should be stored as hypermedia documents and made available by an elaborate set of indexes, among them issues, concepts, and schemes. To enable automatic retrieval by plan fragments for the problem situation or the planned solution, we further suggested to describe plan fragments by the types and numbers of objects contained and certain relations between these objects. While the former descriptors (e.g. issue and concept) have to be attached manually, the latter ones should be automatically inferred. The suggested set of descriptors would allow multi-purpose retrieval.
The vision is, that in the long run cases are derived from GIS plans and the CBR support is an integrated tool in a GIS working environment.

6 NOTES

1 This research was supported by the German Ministry for Education and Research (BMBF) within the joint project FABEL under contract no. 01IW104. Project partners in FABEL are German National Research Center for Computer Science (GMD), Sankt Augustin, BSR Consulting GmbH, Muenchen, Technical University of Dresden, HTWK Leipzig, University of Freiburg, and University of Karlsruhe.

7 REFERENCES


