Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte
Band: 68 (1993)

Artikel: Design reasoning with cases and intelligent objects
Autor: Schmitt, Gerhard
DOI: https://doi.org/10.5169/seals-51844

Design Reasoning with Cases and Intelligent Objects
Conception rationnelle avec des cas spécifiques et objets intelligents
Entwurfsprozess mittels Fallstudien und intelligenten Objekten

Gerhard SCHMITT
Professor
ETH
Zurich, Switzerland

Born 1953, M.Arch (1980) in Architecture, Univ. of California at Berkeley. PhD (1983) at the Technical Univ. of Munich. 1984 - 1988 computer-aided architectural design teaching and research at Carnegie Mellon Univ. Since 1988 Chair for Architecture and CAAD at ETH Zurich. Research interests: Intelligent Design Support Systems.

SUMMARY
This paper presents two new and promising developments in engineering and architecture design paradigms: designing with cases and with intelligent objects. Case-based design (CBD) derives new solutions by adapting existing designs. CBD avoids the problems of extensive search and combinatorial explosion. It helps to maintain the trade-offs and the quality of the case selected for adaptation. For case combination, which should be able to support innovative design tasks, intelligent objects are necessary. They are autonomous design agents of varying complexity which possess first principles knowledge, common sense knowledge, and interaction knowledge, among others. The paper describes experiences with CBD and expected contributions of intelligent objects.

RÉSUMÉ
Cette contribution présente deux développements nouveaux et prometteurs relatifs au paradigme de l'ingénierie et de l'architecture: concevoir à l'aide d'études de cas et d'objets intelligents. Case-based design (CBD) crée de nouvelles solutions à partir de solutions existantes tout en évitant recherche extensive et explosion combinatoire. CBD contribue à maintenir la pertinence et la qualité du cas retenu pour être adapté. Pour des combinaisons de cas capables d'alimenter des tâches innovatrices de conception, de complexité variable et possédant divers degrés de connaissance, entre autres, connaissance de base, de bon sens, des interactions. Cette contribution décrit des expériences avec CBD et de possibles apports des objets intelligents.

ZUSAMMENFASSUNG
Mit Fall-basiertem Entwerfen und dem Arbeiten mit intelligenten Objekten werden zwei Methoden vorgestellt, die den Entwurfsprozess von Architekten und Ingenieuren in der Zukunft verändern werden. Fall-basiertes Entwerfen konzentriert sich auf die Adaptierung guter bestehender Lösungen und vermeidet so viele Probleme traditioneller Computermethoden wie kombinatorischer Explosion und Erzeugung zu vieler Alternativen. Es unterstützt am besten Routine-Design Aufgaben. Das Arbeiten mit intelligenten Objekten erlaubt die Kombination von Fällen oder deren Teile und ist so für innovative Entwurfsaufgaben geeignet.

1. INTRODUCTION

Design reasoning with and based on *cases* has a long history in engineering and architecture. While experienced designers can rely on their own case base accumulated over the years, inexperienced designers rely more on generative methods and on external case bases [1]. Most architecture students learn to design using cases. Magazines present cases in varying depth, from images to detailed descriptions of the processes and the costs involved [11]. In graduate education, case studies are an effective teaching and learning tool. Cases help us to understand a building or a structure as a whole with all inherent trade-offs. They are the final and complex result of a successful design process. They do not necessarily reveal causal relations between design decisions and built results but they may lead to the discovery of such relations. Therefore, they are a very interesting topic of exploration in design computing.

Case-based Reasoning (CBR) and Case-based Design (CBD) are recently established research directions in Computer Science and Artificial Intelligence. They are the latest developments in a series of knowledge acquisition, knowledge representation and reasoning schemes which started with *prototypes*. The next logical step is the development of *intelligent objects* in a design environment which supports the previous representations. *Prototypes* are efficient for the solution of routine design problems, but are less useful for new or unexpected design problems because they have limited automatic adaptation capabilities. They require the ad-hoc definition of all relevant parameters. *Case-based representation* ideally stores all case-relevant information, including structured and unstructured data. It touches on all aspects of existing design and executes parameterization during the adaptation and combination process. Case input, case selection, and reasoning with real and complex design cases still pose formidable computational problems. CBD systems are already under development, *intelligent objects* will combine the advantages of the previous representations with the capability to automatically detect, for example, obstacles and possible violations of building codes and to react accordingly.

2. DESIGN REASONING

Design reasoning is the art of arriving at a design solution from an ill-defined problem specification. The final design solution contains trade-offs. Attempts in the past to see design merely as a decomposition problem or optimization process have failed, although support for this direction still exists [7]. The generation of design with grammars of shapes have been successfully implemented for the reconstruction of historic architecture but have not been widely accepted in practice.

One major reason for the failure of all previous attempts in design automation is a crucial property of human intelligence: the ability to maintain inconsistent solutions and partial solutions in parallel, judging involved trade-offs, while moving towards the design goal. No computer based method so far has offered a solution to this representational and reasoning problem. Researchers are able to analyze complete designs and give an explanation for a possible creation. However these activities have more to do with reverse engineering than with engineering.

Our recent research therefore focuses on a radically different direction. Using complete and existing design solutions and adapting them to a new problem appears promising if the following conditions are fulfilled:

- The new problem has a corresponding solution in an existing case. This requires a sizeable case base and appropriate access mechanisms.
- Appropriate abstraction and representation are available. Those include geometry, which is contained in the CAD model; functional and spatial relations, which can be expressed in a graph; structure, which can be expressed as a finite element representation in combination with geometric and graph representations; performance, which can be expressed as algorithms and constraints. All of these abstractions and representations symbolize important aspects of a completed design but are not necessarily compatible. Moving from one representation to another usually involves loss of information. Abstractions serve the purpose of allowing modifications at a high level without having to concentrate on details. This requires, in order to

guarantee an overall satisfactory solution, the final inspection of the result at the lowest possible level of abstraction, that is the geometric model. At the same time, working on the geometric model alone encourages most designers to focus too much on detail and may lead to a loss of overview and freedom for radical change.

- Appropriate adaptation mechanisms are available. This requires the presence of (i) dimensional adaptation which adapts the geometry of an object and of (ii) topological adaptation which is necessary when dimensional adaptation is unsuccessful.
- Appropriate evaluation methods are available. Every result of adaptation must be evaluated against the goal of the design process. The evaluation indicates whether a solution is successful or needs to undergo a new cycle of adaptation.
- Appropriate visual inspection techniques are available. Every representation of a design is an abstraction which is understood best by only one sector of the design team. The one common representation is the geometric appearance as the result of the design process. Therefore, every product of adaptation must be inspected visually to discover problems that can not be expressed in any of the other representations. For example, a circulation scheme which seems perfectly reasonable in diagrammatic form, might result in completely unsatisfactory architectural solutions.

This particular view represents *one* design reasoning paradigm. It needs extensive knowledge and infrastructure resources but promises complete and functioning solutions. Design reasoning using more traditional approaches such as generate-and-test will produce results that are less predictable and which may not be complete.

3. FROM PROTOTYPES TO INTELLIGENT OBJECTS

The scope of this paper is limited to the detailed discussion of case-based design and intelligent objects. In order to describe their capabilities and the differences to previously developed methods, brief definitions might be helpful:

3.1. Prototypes

Prototypes have been explored as a structure for representing design knowledge [4]. Prototypes are objects for which all parameters are known and pre-defined. Relations between parameters are expressed as rules or similar control structures within the object. Parameters and relations are subject to constraints. Frames are a feasible representation for prototypes.

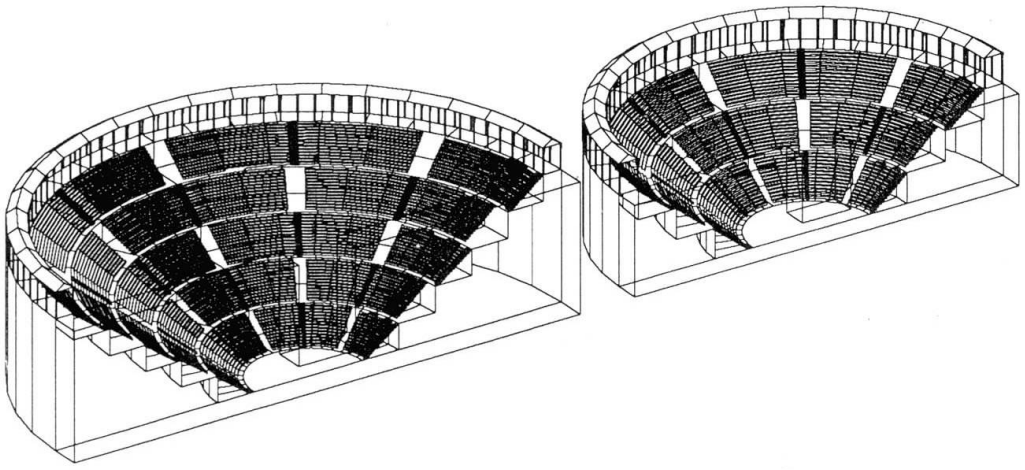

Fig. 1. Examples for objects created with prototypes. Automatic generation of a Roman theatre. Note the different number of seats which correspond to the different number of exits.

An advantage of prototypes is the rapid generation of complete alternatives that are guaranteed to satisfy expected performance criteria. A disadvantage is that the results are constrained and predictable. Therefore, prototypes are most useful for routine design rather than for innovative design.

3.2. Case-based Reasoning (CBR)

Case-based Reasoning is an effective knowledge acquisition and representation scheme for producing complete solutions with minimal search. It is being used increasingly in planning [5] as well as in industrial and military applications. Cases are stored as completely as possible in a case base. Cases are indexed according to their most relevant attributes. Most case-based reasoning applications focus on the activities of case selection and case indexing [8]. Case adaptation is a major concern for design applications.

An advantage of case-based reasoning is that cases need not be parameterized before use as is the case with parameterized objects or prototypes. The parameterization can occur during the reasoning process. Solutions are found quickly if a new problem has a near match in an existing case base or library. The disadvantages are similar to those of prototypes in that limitations exist for the generation of truly new solutions.

3.3. Case-based Design (CBD)

Case-based Design is the application of CBR techniques to design. As designs consist of complex objects, design cases can reach considerable size. Cases may contain, among other things, abstractions and representations of geometry, function, performance, circulation, and the structural system. Adaptation of complex cases is the major challenge in CBD.

Using prior cases is a common strategy for solving complex problems. CBD is different than the traditional rule-based approach for knowledge engineering in that it does not attempt to generalize and compile the knowledge before the problem is specified. Cases are regarded as storage of previous efforts and can be used as short-cuts for finding solutions. Applying this concept to computational processes, means using a given case as the initial state of the search process and searching through only the variations with great similarity to the initial case. The process is divided into case selection and case adaptation, each of which can be solved independently. If the initial case is properly selected, the search for a solution can be carried out much more efficiently. The separation of case selection and case adaptation also suggests a way of compromising that allows the introduction of manual guidance as the form of case selection to reduce the difficulty of machine computations. The implementation challenges of this approach are (i) the capability of recognizing potential solutions, if cases are to be selected automatically, and (ii) the capability to analyze the case and to derive case specific knowledge, for example what is good in this case and should be preserved during the adaptation or what is negative in this case and should be fixed. The final challenge is (iii) to modify the case so that the desirable features are not disturbed and at the same time variations that might solve the new problem may be generated efficiently.

The advantages of case-based design are similar to those of CBR. In addition, it is possible to integrate concerns related to the different representations and to apply adaptation mechanisms to the entire case, thus avoiding problems of combinatorial explosion or endless evaluation loops. Disadvantages are the difficulty of expressing design semantics and the problems of CBD in finding truly new solutions.

3.4. Intelligent Objects

Intelligent objects are autonomous agents that are able to show some kind of intelligent behavior. They can be structured and represented in a way similar to prototypes. In addition, they must "know" about the world surrounding them and thus contain some common sense knowledge. Intelligent objects can also contain first principles knowledge such as the laws of gravity. They are particularly suited for modeling in virtual reality environments where the constraints of two-dimensional screen interaction are absent.

The advantage of intelligent objects is their capability to perform meaningful configuration and simple design tasks without extensive human input. Because they contain common sense design knowledge,

they can help avoid most of the frequently repeated mistakes in design. Disadvantages are their completeness and the difficulties involved in modifying their behavior. They may appear as pre-packaged "black boxes" whose behavior must be accepted without the possibility of critique or change by the user.

3.5. Prototypes, Cases and Intelligent Objects: A Comparison

The use of prototypes and cases have a solid base in architectural education and practice. For computer implementation, the processes involved in reasoning with prototypes and cases must be known and formalized. If successful and fully understood, this will logically lead to the definition of intelligent objects which offer more design freedom while guaranteeing design quality.

Prototypes are most useful when all relevant parameters are known and the design problem is limited in scope, such as the reconstruction of historic architecture [9]. Roman theaters and temples can be reconstructed using a small set of parameters whose relations are known. This enables a programmer to build a tool that will correctly reconstruct an instance of Roman architecture if the appropriate parameters are input. Prototypes thus offer the fastest and "safest" results for well-defined routine design tasks.

In *CBD*, the selection of existing cases as a generating base for new solutions pre-defines the results as well. The reason is the necessary "closeness" of a new solution to an existing case without which the quality of the original case cannot be guaranteed. CBD is most appropriate for design in a sensitive context.

Intelligent objects do not require the extensive adaptation of entire design cases. Instead, they rely on the meaningful combination of known functional or physical design elements. Consistency and overall quality of the resulting design are only guaranteed by the knowledge inherent in the intelligent objects and by the guidance and combination process of the designer. Intelligent objects are a necessary prerequisite for successful case combination which is useful for innovative and creative design.

4. DESIGNING WITH CASES - CASE ADAPTATION

Since 1989 we have developed several case-based procedures for engineering and architectural design. After testing them with practical applications, we believe we have found a valid approach which can be applied in design and related fields. It consists of the following steps:

- Case and site description. This is the necessary first step in which the case and the environment are described to the machine. For this purpose, we developed the pre-processor Mod4 to model the case by inserting walls, doors, windows, rooms and contextual elements such as parcel lines, roads, buildings, lakes, and parks. Mod4 converts the input to AutoCAD format.

- Deriving knowledge from the case. The case input with Mod4 produces additional knowledge which is not visible in the graphical representation but which is present in the model. The system automatically identifies topological relationships among spaces by analyzing the model and its labels. Labels and positions of walls, doors, and windows are used to deduce required spatial adjacencies. Thus we build the different abstractions of a building into a case description which is used for later reasoning.

- Input of new design requirements. If a case must be adapted to programmatic changes as well as to a new context, the following steps are necessary: Specify the maximum and minimum sizes, areas and proportions of spaces (dimensional adaptation). If new spaces are to be added, specify the required adjacencies to other spaces. Indicate the search depth by defining how many rooms are to be reallocated (topological changes).

- Case insertion. The existing case is inserted into a new environment, most likely a new site. The insertion into a new site takes place according to a knowledge base that defines the positive or negative weights of direct links between spaces and contextual elements. For example, a living room facing south, a park, or a yard are positive weighting factors, whereas a bedroom facing a road or other buildings carries a negative weighting factor. The case is checked against its original site first. If a positive link is found, the weight is doubled, if a negative link is found, the negative weight is considered not so significant and thus is divided by two. The insertion is done by maximizing the sum of all links between the new site and the insertion of the case by rotating and mirroring the case.

- Dimensional adaptation. If the insertion satisfies the internal and external requirements, but not the dimensions of the new site, the first step is to dimensionally adapt the existing case. For this purpose, an evaluation detects discrepancies with the new site and converts these conflicts into parameters. The number of parameters can be large in the beginning and must be reduced to produce practical results. The process of dimensionality reduction [6] helps to reduce the parameters to a manageable number and also allows the integration of other design considerations, such as the structural system. If the dimensional adaptation is unsuccessful or produces visually unsatisfactory results, the topological adaptation process begins.

- Topological adaptation. We have tested different approaches for solving this problem. The first and most natural attempt was the use of case-specific rules. These rules fired whenever a conflict between the existing situation and the target situation was detected. As could be expected, the number of rules that were needed grew rapidly and degraded the overall performance of the system. A second attempt was the use of grammar of shapes and the application of neural networks to recognize changes in the grammar. The third and most promising approach employs the wall representation algorithms developed by Flemming. A derivation of Flemming's wall-representation is applied to search for topological variations [2], [3]. Linear programming is used to check the dimensional feasibility of each configuration. The case is used as the starting configuration. The process first removes the number of spaces that are to be reallocated, which already creates some alternatives. Based on these alternatives, spaces are inserted back in one after the other. If one configuration is found to be not acceptable, the entire search branch is discarded. If more than one positive solution is found, the user is prompted to select one to insert it into the new site.

  With this approach, we could solve some of the problems of specialized layout programs, such as the limitation to about ten configurable objects in a fixed layout. Rather than starting without a configuration, we take the existing case as a starting point for a re-configuration and can thus handle a larger number of objects. For example, the time required to solve a 30 space layout problem resulting from a topological change was about five minutes of CPU time on a Sparc Station 10. The algorithms produce a new layout which is then subject to evaluation and three-dimensional visual inspection.

- Evaluation and visual inspection. After the topological adaptation, the geometric results are displayed and evaluated. A visual evaluation of the geometric model is the last step. If the visual evaluation proves unsatisfactory, the above process is re-entered.

5. DESIGNING WITH INTELLIGENT OBJECTS - CASE COMBINATION

Case combination is necessary to achieve truly innovative and creative design solutions. The case combination system under study will be able to combine parts from different cases into a new, complete design. To achieve this goal, the individual parts must possess different types of knowledge in order to facilitate correct combination.

The knowledge is encapsulated in the design objects. In the beginning, there will be physical and functional design objects. *First principles knowledge*, such as the law of gravity, is most important for modeling physical objects. Physical building elements such as walls, beams, columns, and furniture will contain this knowledge. *Common sense knowledge*, which is lacking in most CAD

systems, will also be part of these objects. *Interaction knowledge*, necessary to define object and function interaction, will be encapsulated in physical and functional objects.

Design with intelligent objects must be real-time. This requires fast processors because resolving constraints and complying to rules are computing-intensive tasks. Relatively small intelligent objects, such as furniture and simple structural elements, will therefore be the first implementations. Once the intelligent objects become too complex, such as entire building sections including the corresponding technical installations, system performance will degrade rapidly and interactive use will be impossible. Predictability of the behavior will also be lost.

6. TWO NEW APPLICATIONS

We have applied the approach of case-based design to simple apartment layouts and to two residential buildings by the Swiss architects Campi and Pepsin [10]. Recently, we chose a large office building to test the mechanisms for integrating architectural and structural aspects and to resolve them concurrently. The two following examples will serve to demonstrate new aspects of the CBD approach: Emphasis on the context and learning from the actual process of case-based design.

6.1. Example 1: A house with eight rooms

Figure 2 shows the configuration of three building sites in a given context. The selected case is located on the west site and is to be used, after adaptation, for the two other sites as indicated by the arrows in the figure. However, the spatial needs of the new clients indicate that one more bed room has to be inserted for each adapted case. Therefore, dimensional adaptation will not solve the problem. Instead, after the input of new requirements, the program will generate topological alternatives within the constraints of the new sites.

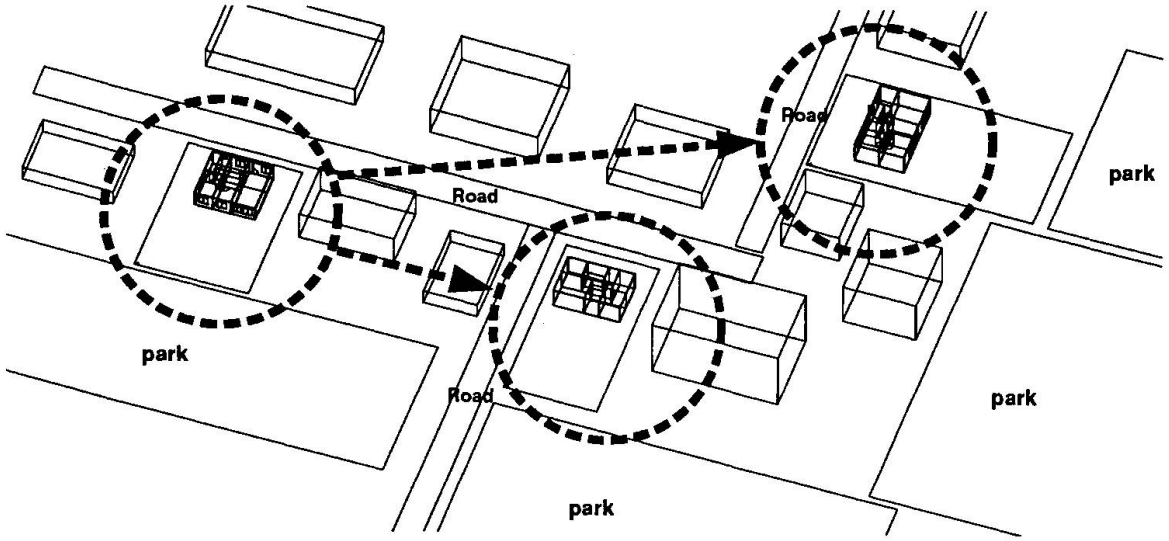

Fig. 2. The site with the original case and two case adaptations

Figure 3 shows the six alternatives for the second site (bottom center of the figure) generated by the program. The search process allows the utility room to be re-located. The reason is that this room has the least links to other spaces and to the exterior. One more bedroom is added. The alternatives are mirrored because there is a road on the west side of the new site: by mirroring the case, negative links between the bedrooms and the road are avoided. For the third site on the east, the same building plans are generated, but they are rotated by 90 degrees. The constraints introduced by the proportion of the site, the access to the road, and the position of the park are stronger than other relations that favor other orientations. In both cases, the user can select any one of the generated alternatives which appear on the screen. After the selection, they are automatically inserted into the site.

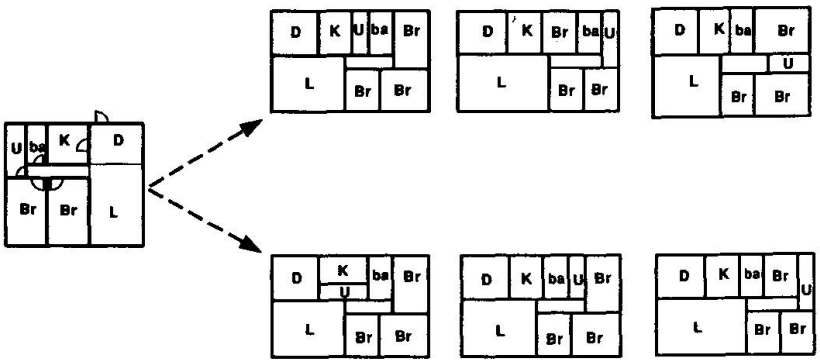

Fig 3. The six alternatives for the new layout with one added bedroom

6.2. Example 2: The timber frame barn

This experiment involved the design and construction of a small timber frame barn in order to detect strengths and shortcomings of the process we developed. The experiment provided a unique opportunity to close the feedback cycle. It consisted of the formulation of a theory, the computer implementation, and the application in practice. Moving from theory to implementation, a few simplifications had to be made, not all aspects of the project could be translated into a computer program. Testing the developed computer approach in practice showed the potential for using intelligent objects. It also revealed missing features in our CBD system.

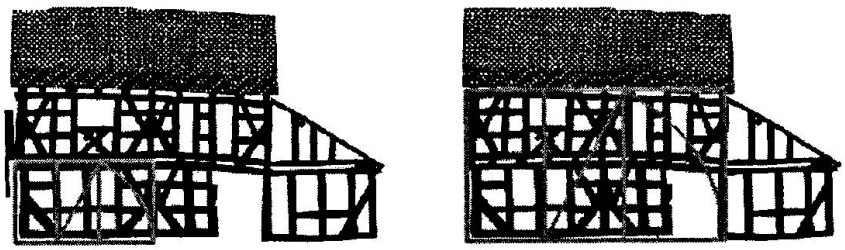

Fig. 4. The selected case and proportional constraints. The facades were topologically adapted to be used for the new case.

The task was to construct a timber frame barn. A number of goals, constraints, and wishes which became formalized only during the design and construction process, guided the design. Given were a program, a site with a 17th century timber frame residential building, and material. Based on these facts, the case-based design process started. The following list describes the procedure, the difficulties encountered in applying our existing CBD system (noted by CBD), and the potential of using intelligent objects in the process (noted by IO):

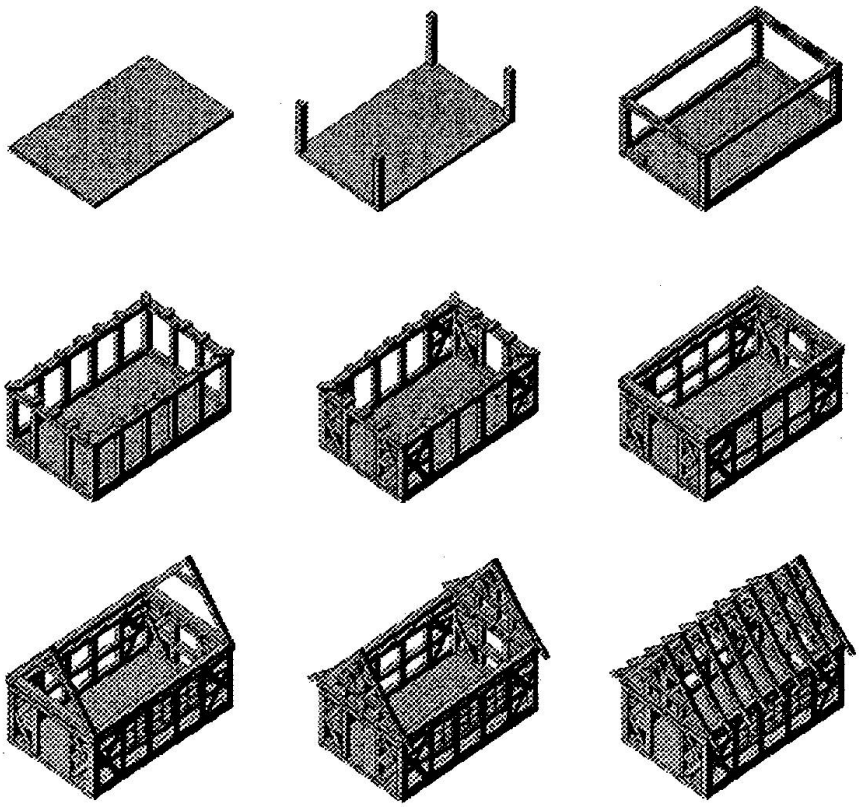

Fig. 5. Adaptation of the traditional construction process, applied to the already dimensionally and topologically adapted case.

- The program. Required was a storage space for a country house. The program resembles most likely that of a small rural barn.
  CBD: Our case base did not contain a building with this program.
  IO: The program will be an object with the appropriate attributes.
- The site. The new building should be close to the existing country house. At the same time, it should not obstruct the main facade of the existing building nor should it be placed too close to the existing structure. By specifying all known constraints, the location of the barn was defined.
  CBD: Parcel lines, existing building, forest, view, and street could be represented with Mod4. So far, we had not yet considered placing a new building on the same site as an existing case, nor had we dealt with non-rectangular sites.
  IO: The barn is an object which operates in an environment defined by geometry, attractive, prohibitive and avoidable features. The barn object has detectors which are sensitive to these features and change the allowed behavior of the object accordingly.
- Material. On site, 17th century oak beams were available with a maximum length of 5.6 m. This strongly suggested the use of the existing material, although some of it was in very bad condition.
  CBD: Our case base did not contain buildings in wood construction. There was no causal link between material and building type for the small scale of the problem at hand.
  IO: Intelligent objects are of little use in this respect, except attaching the request for a certain material as a constraint to the object.
- Construction. Having opted for the material and the site in close proximity of the existing building, traditional construction techniques would be the most reasonable to follow. After inspection of the construction details, it became clear that copying the traditional construction techniques would be too time consuming. Instead, an adaptation was suggested, replacing most of the wooden connections with steel profiles.

CBD: Our case base did not contain construction techniques. It is an important consideration, however, for the reconstruction of historic buildings or for new designs. Thus, a practical problem led to the expansion of the case description in the case base.
IO: Construction techniques and constructability could be made an active attribute of every building element. Existing wooden beams, for example, could request the appropriate fixings. They could disappear from the screen once the span they must cross becomes too large. They could warn the user if they are grossly oversized for the span they cover. They should still warn the user if too much load is added later. In the design of this barn, this would have been an important aid for the design of the roof construction.

- Proportions. All historic buildings in the area are constructed according to well established proportional rules. The common proportion in the elevations are the square and the golden section, the roof gables show angles from 1:1 to 1:2, the most satisfying being $1:\sqrt{2}$. All proportions can be explained by analyzing the design and construction process of the time.
  CBD: Proportional rules are part of our topological adaptation mechanisms.
  IO: design objects could vary within proportional constraints and react positively to desired proportions. For example, 1:1 and 1:2 could be the boundaries, 1:1.414 (square root 2) or 1:1.618 (golden section) could be used as defaults.
- Dimensions. The volume of the barn was as a consequence of the proportional and the material constraints.
  CBD: The existing case adaptation mechanisms are capable of achieving this dimensional adaptation. It reduces the original building basically to one space, which is a topological adaptation of a residence to a primitive hut. Topological adaptation in the other direction would be much more difficult.
  IO: Solving the dimensional constraints will be no problem for intelligent objects.
- Elevations. A direct dimensional adaptation of the existing facade was not possible. A topologically adapted facade passed the evaluation. Symmetry and mirroring were used to guide the topological adaptation process.
  CBD: Our case base did not contain elevations of this kind. However, once the dimensions were decided, the facade can be treated like a floor plan, thus making topological adaptation possible.
  IO: A facade as an intelligent object is a challenge because it has to interact with many other design elements: adjacent rooms, floor slabs, windows, materials, and structure.

The barn was constructed using the computer as the sole construction documentation tool. No paper or other traditional media were used, instead, two on site portables served as constant construction support. The main discovery was that CBD is useful for the building construction process as well.

7. CONCLUSIONS

Case-based design has proven its potential in adapting even complex buildings in theory. In practice, we could demonstrate with an experiment that CBD simulates a human reasoning technique that is found in almost any activity where experience and memory are important. CBD seems to provide a fundamental strategy for avoiding unnecessary re-generation of solutions. In the worst case, it may lead to uninspiring adaptations of existing design. Both the selected case and the adaptation process decide the quality of the final product. There still exist major problems in the representation and manipulation of complete cases which contain structured (e.g., CAD models) and unstructured (e.g., images, user responses, performance data) information.

Intelligent objects are essential elements for the realization of the idea of case combination. Scale and responsibility of the objects are critical parameters in implementation and interaction. Simple objects will be attractive for fast interaction, while complex intelligent objects will have to resolve too many constraints to be used in real-time. Our assumption is, that this limits the practical applicability of intelligent objects to well defined domains. Like cases, intelligent objects contain different representations, each of which is used for a specific design and interaction purpose. Our experiences with CBD have shown that it will be possible to work in real-time also with complex intelligent objects, if we operate on high levels of abstraction, such as diagrams.

8. ACKNOWLEDGEMENTS

The CBD project is funded by the Swiss Nationalfonds: NFP23, Artificial Intelligence and Robotics. I want to thank Boi Faltings, Ian Smith, Kefeng Hua, Simon Bailey, and Shen Guan Shih for the testing and implementation of the ideas presented.

9. REFERENCES

1. AKIN, ÖMER, *The Psychology of Architectural Design*, Pion, London, 1986.
2. COYNE, ROBERT, *ABLOOS: An Evolving Hierarchical Design Framework*, PhD thesis, EDRC 02-15-91, Carnegie Mellon University, Pittsburgh, 1991.
3. FLEMMING, ULRICH, R. F. Coyne, T. J. Glavin, and M. D. Rychener, *A Generative Expert System for the Design of Building Layouts*, Technical Report EDRC-48-08-88, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, 1988.
4. GERO, J., Maher, M.L. and Zhang, W.G. *Chunking Structural Design Knowledge as Prototypes*. The Architectural Computing Unit. Department of Architectural Science, University of Sydney, Australia. January 1988.
5. HAMMOND K., *Case-Based Planning*. Academic Press, Boston, 1989.
6. HUA, K., Smith, I., Faltings, B., Shih, S. and G. Schmitt. "Adaptation of Spatial Design Cases", in John Gero (ed.), *Artificial Intelligence in Design*, Kluwer Academic Publishers, The Netherlands, 1992, pp 559-575.
7. SIMON, HERBERT, *People and computers: their roles in creative design*, Keynote Lecture, Second International Conference on Artificial Intelligence in Design, Carnegie Mellon University, Pittsburgh, USA, June 22, 1992.
8. SCHANK, R. C., *Dynamic Memory: A Theory of Learning in Computers and People*, Cambridge University Press, Cambridge, 1982.
9. SCHMITT, GERHARD, "Rekonstruktion Aventicum", *AutoCAD Special*, Band 2, IWT Verlag, Germany, December 1990, pp. 124-133.
10. SHIH, SHEN-GUAN, "Case-Based Representation and Adaptation in Design", in G. Schmitt (ed.), *CAAD futures '91*, Vieweg, Wiesbaden, 1992, pp. 301-312.
11. "Théâtre de l'Enfance et de la Jeunesse, André Chavanne, Genève", *Werk, Bauen + Wohnen*, 12 Dezember 1992, p. 187.