**Zeitschrift:** IABSE reports = Rapports AIPC = IVBH Berichte

**Band:** 68 (1993)

**Artikel:** Using case-based reasoning for the synthesis of structural systems

**Autor:** Zhang, Dong Mei / Maher, Mary Lou

**DOI:** https://doi.org/10.5169/seals-51850

# Using Case-Based Reasoning for the Synthesis of Structural Systems

Raisonnement rapporté au cas spécifique dans la synthèse des systèmes structuraux

Fallgestütztes Schliessen für die Synthese von Gebäudesystemen

**Dong Mei ZHANG**
Univ. of Sydney
Sydney, Australia

**Mary Lou MAHER**
Univ. of Sydney
Sydney, Australia

## SUMMARY

Design synthesis is defined to be the generation of alternative design solutions. Domain knowledge provides design principles and performance theoretical guidance, design episodes serve as resources in the design synthesis process because they record experience and reasoning steps. The main aim of our research is to explore a design process model incorporating both episode-based design situations and generalized domain knowledge for design synthesis. This paper presents an approach to combining case-based reasoning and decomposition to derive a new design solution by the transformation of previous design situations. The issues of the representation of realistic structural designs in a case base and the transformation of previous design situations are addressed in the paper.

## RÉSUMÉ

On entend par synthèse d'études la génération de solutions de rechange dans l'établissement des projets. Un modèle de processus d'études, en cours d'évaluation, combine un domaine de connaissances, englobant des règles d'études et des caractéristiques de performance, avec des situations d'études épisodiques, dans lesquelles sont mémorisées les expériences et les conclusions de raisonnements déductifs. Les auteurs présentent une méthode qui, à partir d'études mises en archives, permet d'effectuer des analyses et des déductions et, de la sorte, conduit à de nouvelles solutions par transformation de situations d'études précédentes. L'article développe d'une part les questions de la représentation d'études spécifiques de bâtiments et d'autre part, le processus de transformation des situations d'études précédentes.

## ZUSAMMENFASSUNG

Unter Entwurfssynthese wird die Generierung alternativer Lösungen der Entwurfsaufgabe verstanden. Es wird an einem Entwurfsprozessmodell geforscht, das ein Wissensgebiet aus Entwurfsregeln und Leistungsmerkmalen mit episodischen Entwurfssituationen kombiniert, in denen Erfahrungen und Schlussfolgerungen gespeichert sind. Der Beitrag stellt eine Methode vor, wonach aufgrund archivierter Entwürfe Schlüsse und Analysen ermöglicht, und durch Transformation früherer Entwurfssituationen neue Lösungen gefunden werden. Dabei wird auf Fragen der Darstellung realistischer Gebäudeentwürfe in einer Fallsammlung und des Transformationsprozesses näher eingegangen.

## 1 INTRODUCTION

Design is a process in which the experience and knowledge of designers and the design specifications are combined, during which a design description is generated to satisfy the design intentions. In the synthesis of design solutions, alternative configurations are generated and evaluated. During design synthesis, domain knowledge provides design principles and performance theoretical guidance, design episodes serve as resources because they record experience and reasoning steps.

There is no standard method of synthesis suitable for all design problems. The case-based reasoning (CBR) paradigm provides a model for applying prior experience to new problems. It involves retrieving relevant previous cases, adapting the solution from a previous case to solve new problems, and storing the current episode as a new case to be used in the future. CBR as a process model of design synthesis is appealing intuitively because much of design knowledge comes through experience of multiple, individual design situations. For many domains where design knowledge is difficult to acquire and may not be objectively applicable, the case-based paradigm presents a model for the acquisition, organization and reuse of specific design knowledge. Using CBR as a design process model raises the following issues: the identification of what is in a design episode in order to reason about its applicability in a different design context, and the transformation of previous design situations from an original context to a new context.

For design, what is stored in a case reflects the characteristics of design knowledge, as design case retrieval and transformation are based not just on surface features such as the description of design solution, but also on the causal relations between function, behavior, and performance etc. This increases the complexity of the representation and organization of design cases. Whether to include the relational knowledge and governing constraints for a design case within the case or to represent this knowledge outside case memory is still an open research question.

Transformation of a case plays a crucial problem solving role in the CBR paradigm. Transformation includes identifying the difference between the retrieved cases and the new problem and modifying the solution stored in the retrieved case to take those differences into account. The issues raised by transformation are: the representation of domain knowledge about transformation; the maintenance of consistent modification; and the verification of a feasible solution. Previous designs can not be reused without substantial changes. A previous design is either proprietary or customized for a specific context. Proprietary designs (such as Xerox copier) can not be used again without violating laws. Customized designs (such as buildings) can not be used again because the exact context will rarely arise again.

As a result of many efforts toward using CBR for design problem solving, it is found that certain generalized or compiled domain knowledge are essential to address some of the issues in the case-based design model. A hybrid model, therefore, becomes a common approach in many implementations of case-based designs.

In recent years many CBR computer models have used the idea of hybrid systems, and have been developed in engineering design domains. In Wang and Howard's [1988] integrated system for structural engineering design, case-based and rule-based reasonings are combined. A past design can be applied to a similar design problem by replaying its previous design plans. A conventional rule-based module applies design codes and analysis procedures to create the design solutions when case-based design actions are not available. In Faltings et al's [1991] case-based architectural design, the representation of a case involves specific design knowledge and domain dependent knowledge including transformation rules. Case transformation deals with dimensional and topological discrepancies in which a specific design is treated as a starting point of a new design.

As another typical range of case-based designs, some integrated design systems combine model-based reasoning with case-based reasoning. KRITIK [Goel and Chandrasekaran 1989] is an example of this method for the design of small mechanical assemblies. Causal understanding of structure, function and behavior about a device resides in a design model as a functional representation schema, whereas individual mechanical devices are represented an instance of relevant design models. In [Sycara et al 1988], a design case is a graph-based behavior model about a particular device. Case-based reasoning in this approach is viewed as a methodology for selecting and applying various design models rather than specific episodes.

## 2 CADSYN : COMBINING CBR & DECOMPOSITION

CADSYN [Maher and Zhang 1991, 1993] provides a process model for design in which case-based reasoning is combined with a generalized decomposition approach, where the CBR and decomposition approaches complement each other to provide a flexible and comprehensive model of design. In this paper, we will focus on the approach where CBR is adapted to provide a model for selecting and transforming previous design situations to fit a new context using decomposition and constraints knowledge. The process model integrates three distinct types of knowledge: specific design situations, generalized decomposition of a design domain into systems and components, and design constraints. The components of knowledge and main processes for the CBR approach are illustrated in Fig.1.

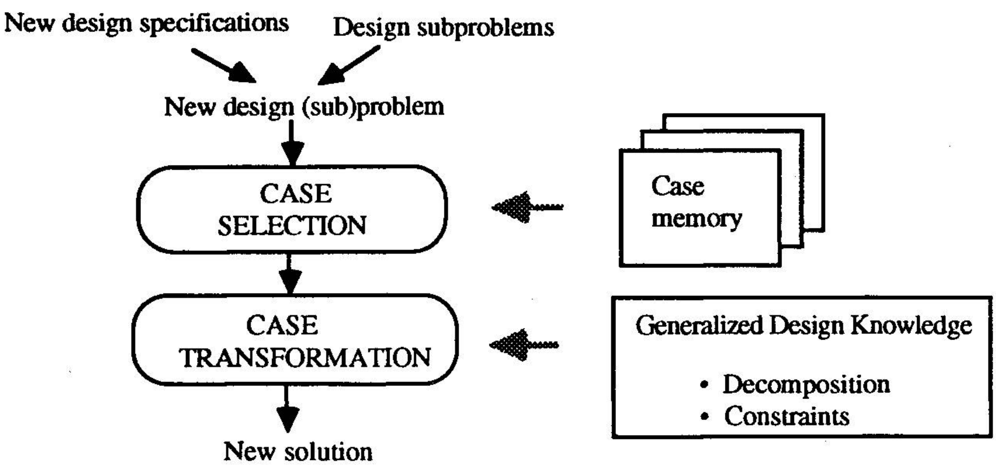



Fig.1 The overall architecture of CADSYN

The problem solving process in CADSYN is primarily divided into case selection and case transformation. Given a new design problem or subproblems, a case or subcase which was designed for a similar context is selected from case memory. The selected case or subcase is then transformed to the new context through modifications which resolve the conflicts caused by difference between the original and the new contexts. A solution, thus, is derived based on (1) the most relevant previous design situation being selected, i.e a close match is found; and (2) transforming the potential solution to fit the new design situation using a domain specific constraint satisfaction approach.

Case selection The selection of the most similar design case consists of two steps, retrieval and selection, as illustrated in Fig.2. The retriever traverses the case memory according to the new problem definition, and identifies the similarities between cases and the new problem. The selector then compares the similar design cases to choose the most relevant one. Case selection involves assessing not only how close the past cases are to the new problem, but also the relative importance of the relevant similarities and differences. To model this selection process, a weighted count of matching features is applied.

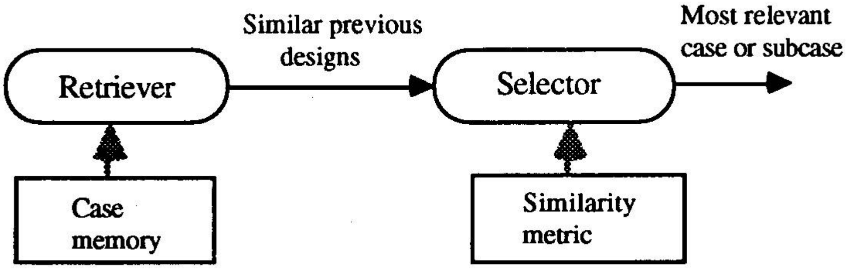



**Fig.2** Design case selection

Case transformation Transformation in CADSYN forms the essence of design synthesis, using a holistic approach to design by starting with a solution and adapting it to fit a new context. Transformation in our model assumes that case selection provides a description of a specific design solution that is close to the acceptable final solution, and transforms those aspects of the design solution that are inconsistent. Transformation can be divided into three logical phases: adapt, verify, and repair, as shown in Fig.3. First, a potential solution to a new problem is proposed as the solution from the selected cases. This potential solution is adapted to change the difference in specifications and the design description, which introduces some inconsistencies between the design specifications and the design description. This step is then followed by verification, the process of evaluating the new solution by checking design constraints, and modification, the process of fixing an inconsistent design to satisfy the violated constraints. Once the solution has been revised, it is verified again. This process proceeds until all constraints are satisfied and a feasible solution is found.

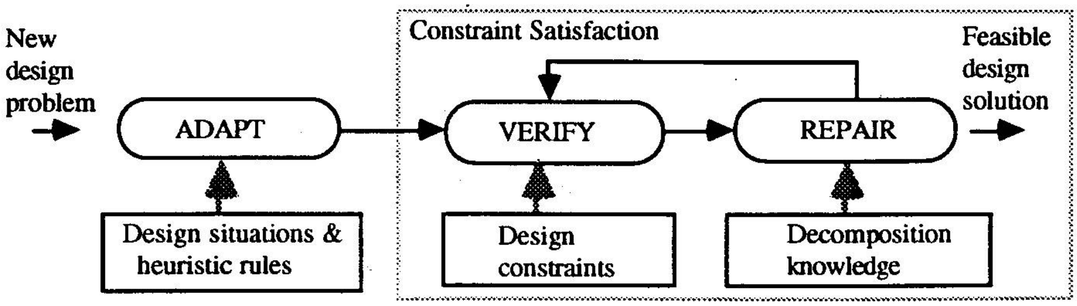



**Fig.3** Design case transformation

To explore the case-based reasoning approach for design synthesis, CADSYN addresses two major issues: the representation of design cases and the transformation of a design case. A design case is a description of the design problem and design solution, in which there are no causal relations to support a design decision. The transformation processuses generalized decomposition knowledge and design constraints as the causal relations that justify and verify design decisions.

## 3 REPRESENTATION OF DESIGN CASES

The formulation of design cases is a prerequisite for a CBR approach to design. In our project, design cases are collected with the cooperation of Acer Wargon Chapman Associates, an engineering consulting firm in Sydney. We were given access to the drawings for the structural design of their building projects. For each building project, there is a set of drawings produced for documentation purposes, primarily for the purpose of communicating the information needed to construct the building. This means, for example, it can be seen on the drawings how much reinforcement each beam has, however, no or very little design information such as lateral load resistance or system design reside on the drawings. To acquire design cases, we augmented the information found on the drawings with interviews with the engineers involved the project. We chose not to put drawings in case memory but to capture the essential design information.

In CADSYN, a design case is represented in a case hierarchy in the form of attribute-value pairs. A design case consists of a supercase part and multiple levels of subcases. This case representation provides the process model with a means to use subcases independently of the entire case. The supercase of a case provides the overall design episode context and general description. Each subcase describes the local context and the solution of a design subsystem. Subcases are indexed individually along with links that can be used to construct the whole case.

As both specific design cases and generalized decomposition knowledge are incorporated to derive a new design solution, a correspondence between them is established as follows: the design description of a particular design case is associated with a set of subsystems from the decomposition knowledge, where each subcase matches a generalized subsystem. This ensures that subsystems and their attributes can be recognized during case transformation for constraint checking and repair.

A design case in CADSYN is the description of a design context comprised of design specifications and a design solution, in which there are no causal relations to support a design decision. In structural design, the content of a design case is constructed in three layers: problem specifications as a global context; a grid representation for each function of building as the geometric context; and structural systems as a design solution for each grid level. The problem specifications of a design case include general architectural specifications and loading information, such as the number of stories, the intended use of the building, etc. The grid representation contains bay numbers and sizes in the two principle orthogonal directions, and other functional and geometric information. A design case of a particular building is illustrated in Fig.4. This office provides four functional spaces: parking, retail, office, and service-core. GEN-CASE shows a overall problem description of the building. The local design context for the office space is shown in the GRID3, and the structural design description associated with GRID3 is illustrated in terms of lateral-systems, gravity-system, transfer etc. The attributes in the structural design description are categorized as requirements (req) and design decisions (des).

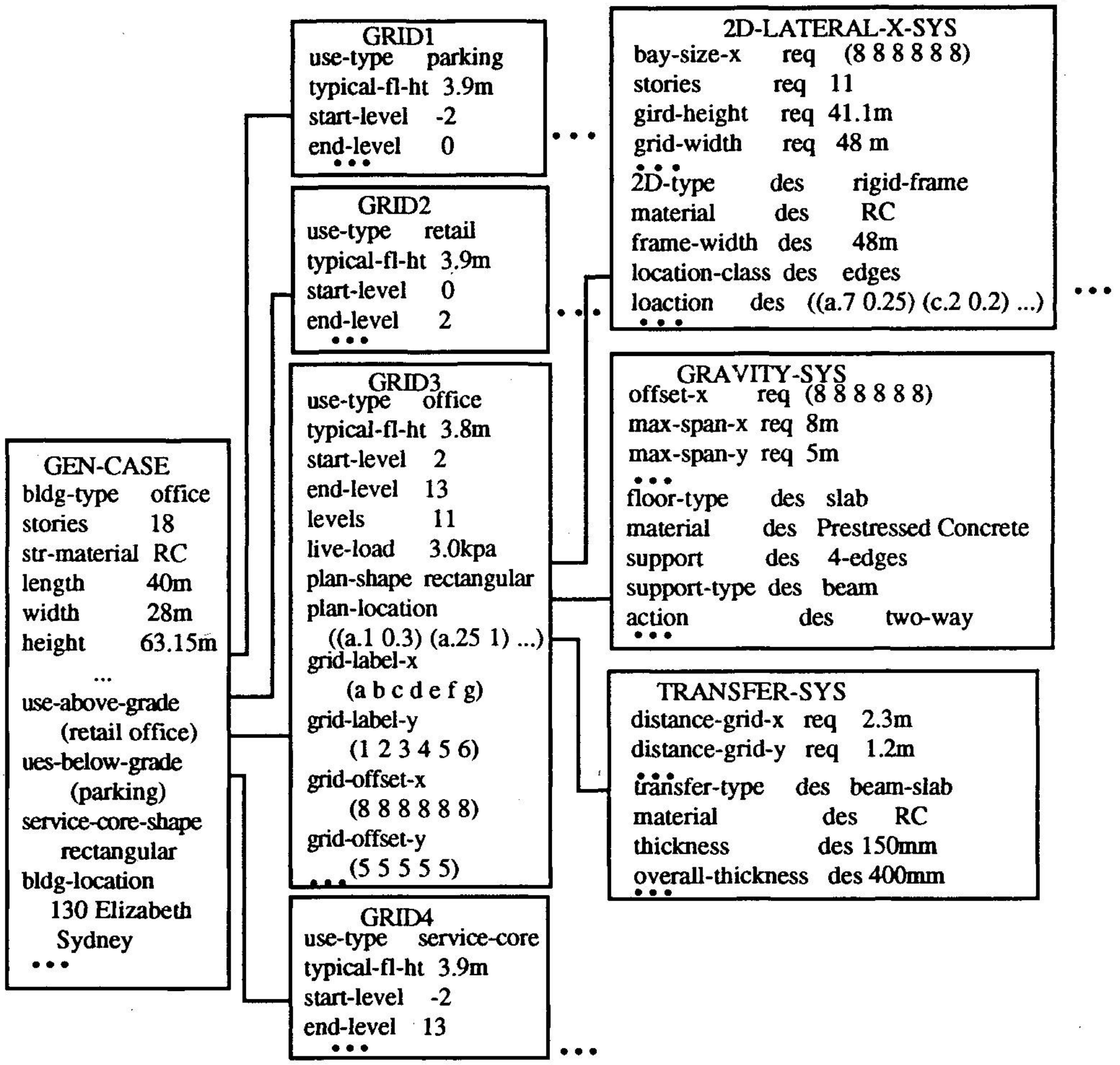



**Fig.4** A partial description of a design case for an office building

## 4 TRANSFORMATION PROCESS IN CADSYN

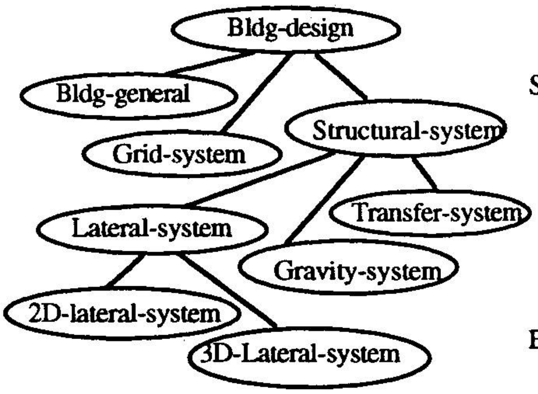
The transformation process of a design case in CADSYN is an adaptation-verification-modification iterative process using constraint satisfaction and decomposition knowledge. In the transformation, the previous design examples act as a starting point for a new design generation and suggests a potential solution to the new problem. An initial solution is firstly constructed by structurally transforming the solution of the most relevant design case. The transformer then adopts a constraint satisfaction approach to check the feasibility of the solution and repair invalid design decisions in the adapted solution.

In this section, the generalized design knowledge used for supporting the transformation process is represented and the strategy for the constraint satisfaction approach in the transformation is described.

## 4.1 Representation of Generalized Knowledge

The transformation process in CADSYN uses two types of generalized design knowledge: decomposition knowledge and design constraints based on the representation of knowledge in EDESYN [Maher 1989].

Generalized decomposition knowledge provides the transformation process with the search space of possible alternatives for design attributes. The decomposition knowledge describes how a design system is to be decomposed into attributes. Each attribute can be defined in one of three ways: further decomposition through subsystem design; selecting from an enumerated set of discrete values; or the evaluation of a procedural function. A generalized system for the gravity system is as shown in the right hand part of Fig.5, where "floor-type", "material" etc. are selected from sets of discrete values, and "typical-span" with computed values by the procedure "get-span".




```
SYSTEM Gravity-system
  ...
  typical-span CALCULATION (get-span)
  floor-type   SELECTION (flat-plate flat-slab ...)
  material     SELECTION (steel rein-concrete ...)
  support      SELECTION (cols-only 2-edges ...)
  action       SELECTION (1-way 2-way)
  ...
END-SYSTEM
```

**Fig.5** A hierarchy of structural subsystems and the description of the gravity system

Fig.5 illustrates a hierarchy of subsystems for a decomposition of the structural design of buildings. The nodes in the hierarchy represent decomposition systems. At the top level, the bldg-design system is broken into three subsystems, namely, Bldg-general, Grid-system and Structural-system. The structural-system leads the synthesis process to a further decomposition of the structural design solution.

Design constraints are used to identify legal decisions and test a potential solution to a new problem. Each constraint is a declarative statement which eliminates a design alternative. For the purpose of design synthesis, it is appropriate to represent constraints as infeasible combinations of attribute-value pairs or relations, since the role of constraints in the early stages of design is to prune the potentially large number of design alternatives.

Examples of constraints for the structural design of buildings are given in Fig.6. CONSTRAINT-1 specifies that the flat-plate and flat-slab are not used as floor types in heavy load buildings such as an office, parking or institution. CONSTRAINT-2 indicates whether the flat-plate and flat-slab work as one-way or two-way given their typical-span.

| CONSTRAINT-1 | CONSTRAINT-2 |
|---|---|
| (bldg-type) in (office parking institution) | (floor-type) in (flat-plate flat-slab) |
| (floor-type) in (flat-plate flat-slab) | (action) = one-way |
| | (typical-span) < 10m |

in : one-of
< : less than
= : equal to

Fig.6 Examples of elimination design constraints

## 4.2 Transformation by Constraint Satisfaction

The transformation of cases in CADSYN can be modeled as a constraint satisfaction problem (CSP), where decomposition knowledge defines the domain of possible values associated with each design attribute; and design constraints represent compatibility and selection knowledge. By treating transformation as a CSP, our method takes an initial inconsistent assignment for the design attributes and incrementally repairs constraint violations until a consistent assignment is achieved.

If the potential solution adapted from the previously selected design case violates design constraints, it will be input to the constraint satisfaction process as an inconsistent assignment for the design attributes. The transformation is characterized by searching for a consistent assignment for all design attributes subject to design constraints. A constraint satisfaction process, as illustrated in Fig.7, is applied to find a consistent solution.

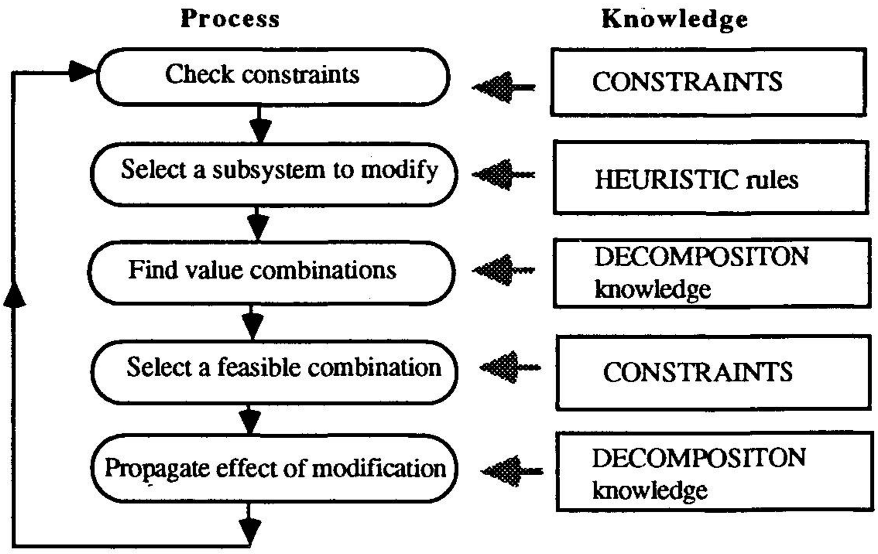



Fig.7 Constraint satisfaction in CADSYN

*Check constraints.* The potential solution provides a set of attribute assignments based on the adaptation of the retrieved case or subcase. This set of attribute assignments is compared to the constraints in the generalized knowledge base to identify violated constraints and the subsystems associated with the constraints. If no constraints are violated, a feasible solution has been found.

*Select a subsystem.* An appropriate subsystem is then identified to be focused on based on a set of heuristic rules. During repairing, the strategy is to fix lower level design decisions rather than change subsystems.

*Find value combinations.* A value for an attribute can be selected from a discrete set of values or it can be computed using a procedure. The possible value combinations are generated by assigning possible discrete attributes, then attributes with computed values are assigned new values by applying relevant procedures based on the value combinations of discrete attributes. The final result of the constraint satisfaction process is a set of possible value combinations in the selected subsystem.

*Select a feasible combination.* The determination of which value combination is used as a new subsystem description is based on the number of constraints which are satisfied. That is, the value combination which satisfies most constraints is regarded as a new design description for the subsystem.

*Propagate effect of modification.* Once a selected subsystem is modified, all relevant attributes in other systems associated with this subsystem are updated by recomputing corresponding procedures. The process iterates from this point by identifying new constraint violations until all constraints are satisfied.

## 5 CONCLUSIONS

The issues raised in this paper are the result of developing the CADSYN process model, and applying the model to the design of structural systems for tall buildings by collecting cases from an engineering consulting company. Direct collection of real world designs leads to difficulties due to the complexity of building design process and the formulation of the case data from design drawings and interviews with designers. One of major issues is that design information is not on the structural drawings. The drawings are used for documentation of the design and contain overwhelming detail on the data needed to construct the object and very little about how the system works. Another issue is capturing the intent of the design so that its adaptation can be consistent following the original context. In general, we augmented the information found on the drawings with interviews with the engineers involved in the project design. The requirements for the design are regarded as a substitution of the designers' intent. A hierarchical representation is used to represent a building design, each subsystem has an associated functional label, set of requirements, and design decisions.

The transformation of a design case in CADSYN is addressed as a design synthesis process using both specific case knowledge and generalized decomposition knowledge. The knowledge about the behavior of structural systems is represented once as generalized decomposition knowledge, rather than repeating it for each case. The knowledge about detecting the design failures is represented as constraints to verify the adapted design and ensure a consistent modification. A new design is generated by adapting cases and performing a constraint satisfaction process in which decomposition knowledge provides a space of possible alternatives for a design attribute, and elimination constraints represent compatibility and selection knowledge. In the development of a constraint satisfaction approach to transformation, how to accumulate the appropriate generalized knowledge is a major issue as there is very little causal knowledge available at the preliminary design stage. Constraints, for example, are used to determine whether an adapted design is feasible, but as knowledge used for preliminary design, such constraints are based on heuristics rather than on an analysis of the behavior of the solution.

**Acknowledgements**

We would like to thank Acer Wargon Chapman Parters for providing access to their structural design reports and drawings and for participating in discussion. Special thanks to Rita Villamayor for the infinite number of hours she spent looking at the drawings to find some design data. Thanks also to M.Balachandran for his contribution to the development of these ideas.

## REFERENCE

1. Faltings, B., and Hua, K. et al (1991). Case-Based Representation of Architectural Design Knowledge. In Proceedings of the DARPA Workshops on Case-Based Reasoning. Washington, D.C. 307-316.
2. Goel and Chandrasekaran, (1989). Use of Device Models in Adaptation of Design Cases. In Proceedings of the DARPA workshop on Case-Based Reasoning. Pensacola Beach, Florida. 100-109.
3. Maher, M. L. (1989). Synthesis and Evaluation of Preliminary Designs. In J. S. Gero (editor), Artificial Intelligence in Design. Springer-Verlag. 3-14.
4. Maher, M.L. and Zhang, D. M. (1991). CADSYN: Using Case and Decomposition Knowledge for Design Synthesis. In J. S. Gero (editor), Artificial Intelligence in Design' 91, Butterworth-Heinemann, UK. 137-150.
5. Maher, M.L. and Zhang, D.M. (1993). CADSYN: A Case-Based Design Process Model. AI EDAM (to appear).
6. Sycara, K.P. and Navinchandra, D. (1989). Integrated Case-based Reasoning and Qualitative Reasoning in Engineering Design. In J. S. Gero (editor), Artificial Intelligence in Design, Springer-Verlag. 231-250.
7. Wang, J. and Howard, H.C. (1991). A Design-Dependent Approach to Integrated Structural Design. In J. S. Gero (editor),Artificial Intelligence in Design'91. Buttorworth-Heinemann, UK. 151-170