Representation and Reuse of Design Knowledge and Design Rationale for Design Decision-Making

Jie Hu[1], Yinghong Peng[1],
Guangleng Xiong[2]

[1] School of Mechanical Engineering, Shanghai Jiao Tong University,
Shanghai, P.R. China
hujie@sjtu.edu.cn
[2] Department of Automation, Tsinghua University,
Beijing, P.R. China
glxiong@cims.tsinghua.edu.cn

Abstract. Design rationale comprises information from two aspects: information about decision making and information about the artifact itself. This paper presents an approach of representation and reuse of design rationale for design decision-making. The relations between design description and design decision are analyzed and the conditions for consistence are given. The ontologies for design rationale representation are described. The constraints of consistence are deducted based on the conditions and the design rationale model. A system for presentation and reuse of design knowledge and design rationale is developed. The method described in this paper was used to develop a system, which is applied to presentation and reuse of design knowledge and design rationale for a railway vehicle bogie design.

Keywords: Knowledge representation, Design rationale, Ontologies, Consistence, Decision-Making.

1 Introduction

Knowledge about the rationale for a design—how and why a device is designed as it is—can be valuable, but is difficult to capture in reusable form. A design rationale is an explanation of how and why an artifact, or some portion of it, is designed the way it is. A design rationale is a description of the reasoning justifying the resulting design—how structures achieve functions, why particular structures are chosen over alternatives, and what behavior is expected under what operating conditions. In short, a design rationale explains the "why" of a design by describing what the artifact is, what it is supposed to do, and how it got to be designed that way [1].

Design rationale represents design knowledge that is used by people—possibly assisted by interactive computer aids—for engineering tasks including manufacturing, verification, diagnosis, and redesign. In current engineering practice, this knowledge is often lost because designers forget it, leave the project, or are inaccessible in a large organization.

© A. Gelbukh, Á. Kuri (Eds.)
Advances in Artificial Intelligence and Applications
Research in Computer Science 32, 2007, pp. 111–119

Received 16/06/07
Accepted 31/08/07
Final version 21/09/07

The design documentation at present focuses on the data about "how to implement the design artifact", while the information about "why the product is designed as it is" is ignored. These information is believed to be as important as implementation data. It helps to support design reasoning, to facilitate design communication and to further design knowledge accumulation and development so that the quality and efficiency of product design are improved and the product competence of enterprises are enhanced. Design rationale (DR) research is exploring the representation, capture and uses of these information. Explicitly represented rationale can help individual designers clarify their thinking about a design [2-6]. Perhaps more importantly, rationale capture can support design by concurrent engineering (CE) teams over time. The reasoning behind decisions becomes available for all team members to critique and augment. Most existing rationale capture approaches support only individual users and are thus not suited to team contexts.

Representing design rationale requires that one explicitly document the reasoning and argumentation in design [7]. Szykman *et al.* [8] presented two fundamental representations of design rationale. First is the notion of design rationale as the recording of the design intent of an artifact. For example, in traditional mechanical design, rationale might include a functional description, geometric or assembly constraints and performance criteriaThe second view is of design rationale as a record of the design process, the communications among agents, the decision-making that occurs, as well as the decision-making process. This view of rationale has been often studied in the software design and computer-supported collaborative work communities, where one goal is to support organizational intelligence and group decision making.

This paper is to solve the problem of "how to represent and reuse design rationale in collaborative product design" based on a combination of current design rationale research and engineering practices in manufacturing enterprises. The relations between design description and design decision are analyzed and the conditions for consistence are given. A system is developed and applied to presentation and reuse of design knowledge and design rationale for a railway vehicle bogie design.

2 Representation of Design Knowledge and Design Rationale

Multidisciplinary collaborative design is a methodology for the design of complex coupled systems, in which the synergistic effects of coupling between various interacting disciplines are explored. These subsystems are usually geographically distributed and implemented in different computers to support complex design projects carried out in multidisciplinary teams contexts.

Multidisciplinary designers are no longer merely exchanging geometric data, but more general knowledge about design and the product development process, including specifications, design rules, design constraints, and design rationale. Furthermore, this exchange of knowledge often crosses corporate boundaries. As design becomes increasingly knowledge-intensive and collaborative, the need for computational frameworks to enable engineering product development, by effectively supporting the formal representation, capture, retrieval and reuse of product knowledge, becomes more critical.

Design rationale comprises information from two aspects: information about decision making (issues, alternatives, criteria, objectives) and information about the artifact itself (requirements, function, structure, behavior). To support the four kinds of computer services, we build an integrated information model of the two aspects. The model is illustrated in Fig. 1. The concepts and relations between them are explained as follows.

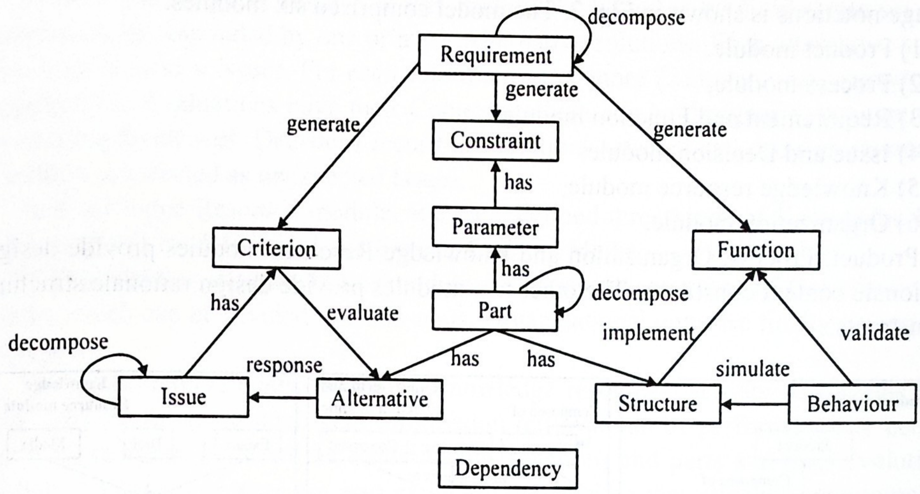

Fig.1. The design rationale model.

Requirement: The specification describing the function, structure and behavior of the design. It can be divided into criterion and constraint.

Criterion: The expectation for measurable product properties, such as cost, manufacturability, safety, etc.

Issue: A decision problem. It involves a choice among a set of alternatives, relevant criteria, and information used to compute the utilities of the alternatives in terms of criteria.

Alternative: The description of design or some part of it. Decisions are choices among alternatives.

Constraint: An expression specifying a limit on the range of values of one or more parameters.

Parameter: A variable describing some property of design artifact. Parameters can describe the function, constraint and behavior of the design.

Part: Elements composing of product design. It can be components, assemblies, or subsystems.

Function: Expected behavior of design artifact in some operation environments.

Structure: The physical and/or logical composition of an artifact, typically in terms of composition of parts, and connection topologies. Structure description is the basis for realizing the artifact and is depicted with schematics or drawings.

Behavior: States and changes of the artifact in some operation environment. Behavior descriptions are relations over quantities of parameters if states are described by these parameters.

Dependency: Determination relations between design element (such as parameters, requirements, constraints and issues). The relations are unidirectional. Dependency management is necessary because design reuse must involve design modifications.

These concepts and relations between them form the core of design rationale model.

The ontologies to describe design rationale model using Unified Modeling Language notations is shown in Fig. 2. The model comprised six modules:

1) Product module,
2) Process module,
3) Requirement and Function module,
4) Issue and Decision module,
5) Knowledge resource module,
6) Organization module.

Product, Process, Organization and Knowledge Resource modules provide design rationale context constructs. The other two modules provide design rationale structure constructs.

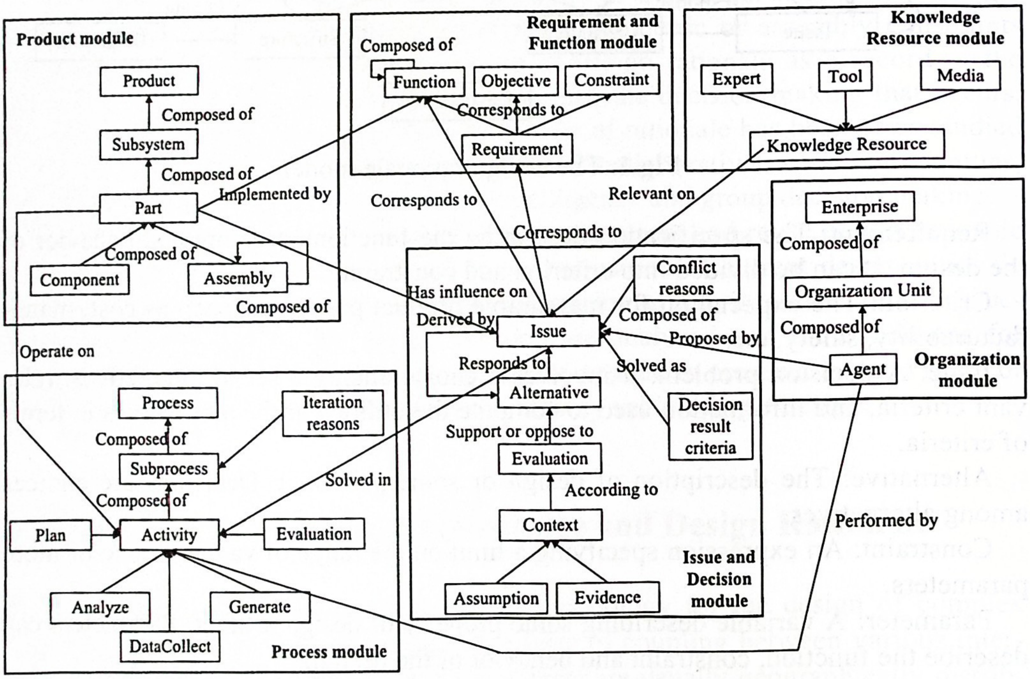

Fig. 2. The ontologies to describe design rationale model.

The objects and relations in the model are explained concisely as follows:

1) Product module. In this module, Products are composed of subsystems. Subsystems are composed of Parts. There are two kinds of Parts: Components are indecomposable and Assemblies are composed of Components and subassemblies.

2) Process module. In this module, Processes are composed of Subprocesses. Subprocesses are composed of Activities and sub-Subprocesses. Activities can not be divided further. We define five kind of Activity: Plan, Analyze, Data Collect, Generate and Evaluate. Iteration is modeled as a kind of Subprocess.

3) Requirement and Function module. In this module, Requirements and Functions can be decomposed to more specific Requirements and Functions. Requirements are distinguished as initial requirements and derived requirements. Initial requirements are explicit statements of customers. Derived requirements are derived from initial requirements during the design process. Functional demands, Objectives, and Constraints can be extracted from Requirements.

4) Issue and Decision module. In this module, An Issue can be divided into several sub-Issues, or responded by one or more Alternative solutions. Each alternative solution may raise new issues. For each solution, one or more Evaluation may support or oppose to it. Evaluations have their Contexts, which can be designer's Assumptions or existing Evidences. Decisions record the solutions selected and decision rationale. Conflicts are treated as unexpected Issues.

5) Knowledge Resource module. We have defined three kinds of Knowledge Resources in this module, which are Expert, Tool and Media.

6) Organization module. In this module, Enterprise is composed of Organizational Units, which can be divided into sub-units. Organizational units are finally composed of Agents.

The parts, requirements, functions, knowledge resources, activity and agents are explicitly modeled, so intelligent information retrieval can be performed. The corresponded relations between the requirements, functions and parts and their evolution during the design process are also explicitly modeled so that effective requirements management can be performed. The multiple views to present design rationale can also be constructed on the information model.

3 Consistent Analysis of Design Rationale for Design Decision-Making

For reuse of design ration, the consistent of design rationale should be analyzed. The consistent analysis algorithm of design rationale can be described as:

```
Procedure Consistence-Checking(Rationale dr)
Begin
/*Decision behaviour checking*/
  /*Constraints for feasibility*/
  For i From 1 to M=total number of alternatives Do
    IF(selected(Ai) And totalEvaluation(Ai)<0) Then
      Print("feasibility constraints violated!",i);
      Break;
  /*Constraints for optimality*/
  For i From 1 to M=total number of issues Do
    For j From 1 To N=total number of alternatives of
                    Issue i Do
      If (selected(Aj) And totalEvaluation(Aj)!
                              =maxTotalEvaluation(Ii)) Then
      Print("optimality constraints violated!",i);
      Break;
/*Decision content checking*/
  /*Constraints for requirements*/
```

```
For i From 1 to M=total number of requirements Do
  IF(decomposed(Ri) Then
    For j From 1 To N=total number of keywords of Ri Do
      If(notContainedIn(Kj,keywords
        (subRequirements(Ri)))) Then
      Print("requirement constraints violated!",i);
    Break;
  IF(generated(Ri) Then
    For j From 1 To N=total number of keywords of Ri Do
      if(notContainedIn(Kj,keywords
        (generatedIssues(Ri)))) Then
      Print("requirement constraints violated!",i);
      Break;
/*Constraints for issues*/
For i From 1 to M=total number of issues Do
  IF(decomposed(Ii) Then
    For j From 1 To N=total number of keywords of Ii Do
      If(notContainedIn(Kj,keywords(subIssues(Ri))))
      Then
      Print("issue constraints violated!",i);
      Break;
  IF(responded(Ii) Then
    For j From 1 To N=total number of keywords of Ii Do
      If(notContainedIn(Kj,keywords
        (respondedAlternatives(Ii)))) Then
      Print("issue constraints violated!",i);
    Break;
      If(notContainedIn(Kj,keywords
        (relatedCriteria(Ii)))) Then
      Print("issue constraints violated!",i);
      Break;
/*Constraints for alternatives*/
For i From 1 to M=total number of alternatives Do
  For h From 1 To L=total number of derived Issues of Ai
  Do flag=0;
    For j From 1 To N=total number of keywords of Ii Do
      If(containedIn(Kj,keywords(derivedIssues(Ai,h))))
      Then flag=1;
      Break;
      If(flag==0) Then
      Print("alternative constraints violated!",i)
      Break;
For i From 1 to M=total number of alternatives Do
  flag=0;
  For j From 1 To N=total number of keywords of Ai Do
    If(containedIn(Kj,keywords(relatedArgument(Ai))))
    Then flag=1;
    Break;
    If(flag==0) Then
    Print("alternative constraints violated!",i)
    Break;
/*Constraints for criteria*/
For i From 1 to M=total number of criteria Do
  flag=0;
```

```
      For j From 1 To N=total number of keywords of Ci Do
        If(containedIn(Kj,keywords(relatedArguments(Ci))))
        Then flag=1;
        Break;
        If(flag==0) Then
        Print("criterion constraints violated!",i)
        Break;
End
```

4 System and Application

We develop a system for presentation and reuse of design rationale. The functionality of the system includes three aspects: design rationale capture, design rationale browsing and design rationale query.

1) The capturing of design rationale should be integrated firmly with design process so that designers could record and index design rationale information while performing design tasks. The tools of requirements management are integrated into the design process support system. The structure of design rationale information is formed during capturing with the support tools. The contexts of design rationale are generated from the design activities. The contents of design rationale are indexed according to domain ontologies by the designers interactively.

2) Design rationale is browsed through multiple views. Each view presents some aspects of the contents, structure and contexts of the design rationale information so that the access to design rationale information resources can be guided.

3) Design rationale can be queried by its properties, contexts and contents. The information can be browsed through proper views after the instances are found by query.

The architecture of the system has three levels:

1) Application level. In this level, the tools of design rationale capturing, browsing and querying are integrated with design process support system. Services are actively provided according to specific design task.

2) Description level. In this level, design rationale information is described based on ontologies. The contexts of design rationale information are described using enterprise ontology, including departments, processes, and activities, etc. The basic properties of design rationale information are described using information ontology, including media, abstract level, etc. The contents of design rationale information are described using domain ontology to specify domain keywords. These descriptions are the basis for intelligent information retrieval.

3) Object level. This level is comprised of several kinds of design rationale information instances. Instance classes include issues, parts, documents, requirements, etc.

Following is an example to illustrate how to application the methods and system to design of a railway vehicle bogie. The bogie in the example is shown in Fig. 3.

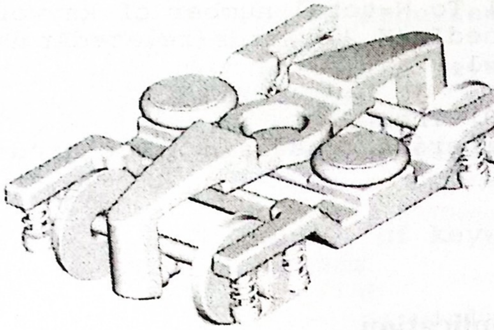

Fig.3. An example of bogie design.

In this example, issues, alternatives, criteria and related document objects are indexed using enterprise ontology, information ontology and domain ontology so that intelligent information retrieval can be performed. The contexts of design rationale information are described using enterprise ontology, including departments, processes, and activities, etc. The basic properties of design rationale information are described using information ontology, including media, abstract level, etc. The contents of design history information are described using domain ontology to specify domain keywords. The design rationale information instance is described as follows.

```
Class:             Issue
Name:              Spring selection
Media:             Text
Abstract level:    Qualitative
Requirement:       Stability, safety
Function:          Load allocation, concussion alleviation,
                   moving stability assurance
Product:           Australian food car
Subsystem:         Bogie
Part:              Spring
Process:           Australian food car development
Activity:          Bogie spring design
Department:        Design department
Designer:          J Zhang
Keywords:          Freight car bogie, spring, elasticity,
                   concussion alleviation
Sub-issue:         None
Super-issue:       Freight car bogies design
Solutions:         Round spring
```

Decision support by multi-attribute utility theory is provided to avoid personal bias. For each decision issue, there are a set of criteria and an objective function to let the designer to select between alternatives. Criteria include aspects such as cost, safety, manufacturability, etc. Each alternative is evaluated against the criteria and given an evaluation point based on the objective function, which makes a trade-off between criteria. Different criterion will have different power in the objective function and the power may change due to different requirements. Facts and assumptions which lead to the evaluation point are also recorded.

5 Conclusions

This paper presents a method of representation, reuse and consistent analysis of design knowledge and design rationale. By describing design rationale information with homogeneous ontologies, heterogeneous engineering information sources can be accessed uniformly. This ontology-based approach overcomes knowledge acquisition bottlenecks of design rationale system construction. The consistence is deducted based on the conditions and the design rationale model. The scheme of representation and consistent analysis is the basis for design decision-making.

Acknowledgement

This paper is supported by the National Natural Science Foundation of China (Nos. 50575142 and 60304015), and the Shanghai Committee of Science and Technology (Nos. 055107048 and 04ZR14081)

References

1. Gruber, T. R. and Russell, D. M.: Design Knowledge and Design Rationale: A Framework for Representation, Capture, and Use. Technical Report KSL 90-45, Stanford University. (1991) 1-40
2. Lee, J. and Lai, K.Y.: What's In Design Rationale?. Human-Computer Interaction 6 (1991) 251–280
3. MacLean, A., Young, R., Bellotti, V., and Moran, T.: Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Computer Interaction 6 (1991) 201–250
4. Klein, M. Supporting Conflict Resolution in Cooperative Design Systems. IEEE Systems Man and Cybernetics 21 (1991) 1379-1390
5. Shum, S. A. Cognitive Analysis of Design Rationale Representation. PhD dissertation. University of York (1991)
6. Lee, J. Design Rationale System: Understanding the Issues. IEEE Expert 12 (1997) 78–85
7. MacLean, A., Young, R., Belloti, V., and Moran, T.: Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Computer Interaction 6 (1991) 201–250
8. Szykman, S., Sriram, R. D. and Regli, W. C.: The Role of Knowledge in Next-generation Product Development Systems. Journal of Computing and Information Science in Engineering 1 (2001) 3–11