Modelling of Product Configuration Design and Management by Using Product Structure Knowledge

Bei Yu and Ken J. MacCallum
University of Strathclyde
CAD Centre, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ, UK, Tel: +44 41 552 4400 Ext. 2374, Fax: 01223-439585. email: bei@cad.strath.ac.uk, ken@cad.strath.ac.uk

Abstract
For whole life cycle of product development, configuration can be considered from two aspects: configuration design which is the activity of creating configuration solutions, and configuration management which is the process of maintaining a consistent configuration under change. Both configuration design and configuration management are complex processes for many products, particularly when the product structure is complex in terms of a large number of elements with different relationships, the configuration problem will be significant.

This paper presents an AI-based system to support configuration design and management. The contribution is of a system which models product configuration knowledge, and uses a Reason Maintenance System as an inference engine to assist the designer to create a product structure in terms of configuration solution. At the same time, the configuration consistency is maintained by the inference engine.

Keywords
configuration design and management, product structures, reason maintenance mechanism, logic-based truth maintenance system, constraint-based reasoning.

1 INTRODUCTION

In engineering design, any product or machine can be viewed as a technical system which consists of elements and their relationships. Configuration thus can be regarded as a process: from a given set of elements, to create an arrangement by defining the relationships between selected elements that satisfies the requirements and constraints. For whole life cycle of product development, configuration can be considered from two aspects: *Configuration Design* and *Configuration Management.* Configuration design is the process of creating configurations, in which it is concerned with the elements selection and the ways of configuring elements. In contrast, configuration management is the process of maintaining a consistent configuration under change, in which it is concerned with the configuration consistency. Especially when the decision of selecting elements is changed, configuration management should trace all the decisions which are related to the changed decision and revise them if necessary to maintain consistency among elements and decisions. Both configuration design and configuration management are complex processes for many products, particularly when the product structure is complex in terms of a large number of components with different relationships, the configuration problem will be significant.

For many industrial companies, the product structures are very complex. Even for a routine design, it can be extremely difficult to configure a new product structure rapidly and correctly. If the product needs innovation,configuration in addition becomes intermixed with other aspects of design.

In practice many products reuse past designs or components. Since most products are changed depending either on their functionality or on particular requirements, the products are renewed incrementally rather than being changed totally to a new one. Reuse and adaptation of previous products is very important in the design process. Adapting established configurations to new requirements, functionalities, or technologies, requires an approach to configuration management rather than design; that is maintaining the consistency of configurations under change, rather than simply selecting.

This paper presents an AI-based system to support configuration design and management. The approach models configuration design and management by formalising configuration knowledge such as product structures, constraints knowledge and configuration decisions. A reason maintenance inference engine is developed for maintaining the consistency between decisions and selected elements based on the proposed configuration knowledge structures.

2 CONFIGURATION WITHIN PRODUCT DEVELOPMENT

In engineering product development, the process of whole life cycle of a product starts from the market or customer requirements, into design specification stage, then through conceptual design, into detail design, and on to manufacturing, eventually to sales phase(Pugh 1991). As a generic design activity, configuration design is viewed as the tasks of determining different relationships, and interdependencies among product elements, design decision and options, so as to form a consistent product structure or model that satisfies all requirements and constraints.

Given a set of requirements and constraints, the configuration process begins by examing the product family which includes all elements, makes decisions on selecting elements from it, and combines these elements into a consistent artifact. However, configuration can be a bottleneck in the design process. The market demands are for short lead times and improved quality of the product. If the configuration space is large, it can take a long time to search, choose and make correct decisions. Complexity of the product family and configuration information also leads to difficulty in the configuration process, particularly in maintaining consistency and dependency with change. Under a time pressure it is easy for designers to change a design feature and overlook a "knock-on" effect of the change.

Most new products are obtained by incremental product development; therefore reusing previous design concepts and design knowledge is an important aspect in the design process. To do this successfully, the designer needs to know which sort of previous concepts and knowledge can be reused, and how they can be applied. The interdependencies of decisions, however, is not always explicit from past designs, making a further source of error(MacCallum 1992).

In addition, the designer must optimise the design carefully, evaluate alternative configurations where possible, and carefully trade off a number of possibly conflicting factors, such as efficiency, cost, complexity, and reuse of existing standard components to reduce design and tooling costs.

Information changes which cannot be known in the beginning also delay the whole product development time by changing elements in the late production stage. If the elements have to be changed for some reasons in the manufacture stage, the configuration process in the design stage would need to be done again. These types of changes are unacceptable in a "right first time" design process.

The key aspect of this problem for which a designer needs help is not only in designing itself, but also in maintaining consistency across design decisions.

3 MODELLING OF PRODUCT STRUCTURE KNOWLEDGE

The overall goal of using the system is to produce a product breakdown structure for a new product which is a legal combination of the selected elements, from a series of design decisions. The configuration knowledge includes product information, constraints, requirements, decisions and configuration solutions. Various types of product structure knowledge are formalised in order to support configuration design and management.

Product Family Classification Trees (PFCT)

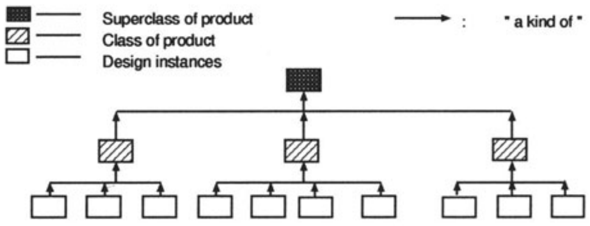

Figure 1 Product Family Classification Tree

A product range can be classified as the *Product Family Classification Trees (PFCTs)* which is the tree structures that present a class of product and its modules from an abstract level to product instances. Each node in the tree represents a product or modules class with its parts breakdown. The ancestor-descendant relationship of two classes is presented as "a kind of", i.e., a class of product is a kind of the superclass of product.

Fig. 1 shows the structure of a general product family classification tree. The links between the levels are represented as "a kind of". For example (see Fig. 2), in the class of the domestic heater, *Heater* is on the top level which is the most abstract concept in the heater products. *Heater* could be classified into three types of heater: *Direct Heater,* *Indirect Heater* and *Direct&Indirect Heater*. The types of heater are classified down to the most specific heater model in terms of the design instances.

There will be several product classification trees which are related to each other. In other words, all existing modules or parts and elements that might be configured can be found in the given knowledge sources in terms of their own Product Family Classification Trees. For example, related to the heater product family, there are two other product classification trees called *Motor* and *Control unit*. They are related to *Heater* classification tree through a parts breakdown structure, i.e., heater consists of motor and control unit.

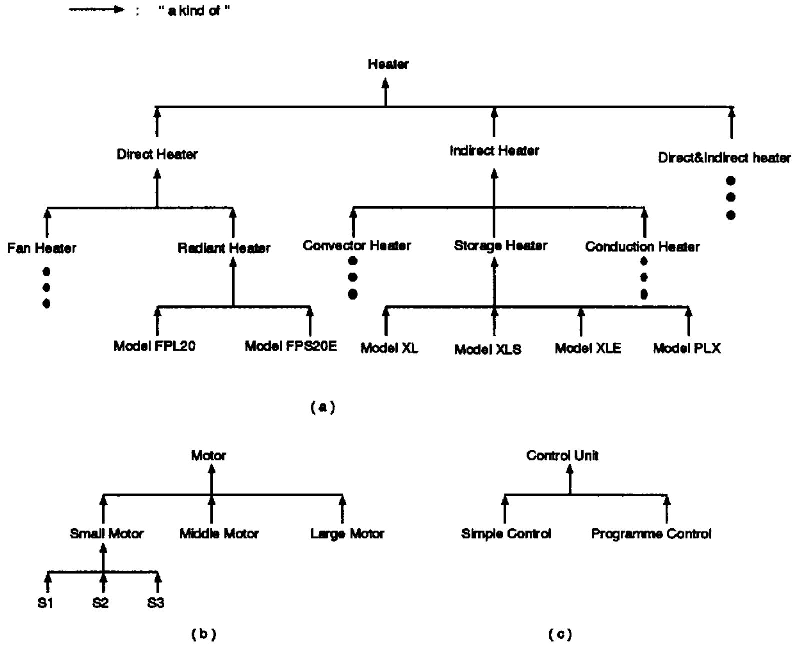

Figure 2 Domestic Heater Classification Tree

Design Constraints Knowledge

This is a set of decision constraints related to subsystems, parts and elements to be selected. These constraints can be separated into two types, one of which presents the dependencies among the Product Family Classification Trees, and the other presents a set of limitations on the possible combinations of subsystems, parts and elements that are feasible in a single design.

As an example of the first type of constraint knowledge in terms of dependencies, *Direct Heater* and *Indirect Heater* which under the *Heater* product family tree need a *Small Motor* that exists in the *Motor* product family tree for their *Motor* part selection. This can be represented as a logical dependency between *Heater* and *Motor* family trees, i.e. if *Direct Heater* or *Indirect Heater* is chosen then *Small Motor* will be chosen for its *Motor* part and vice versa. The second type can be represented as a logical relationship for limiting the possible combinations of subsystems, parts and components. Such restrictions can be represented as AND, OR and NOT relationships, and reduce the possible choices at design time.

The constraints knowledge comes from either the designer who is in charge of doing configuration, or the end user who provides particular requirements. It may reduce choices at decision time.

Product Breakdown Structure (PBS)

As the form of a configuration solution, *Product Breakdown Structure (PBS)* presents a list of elements and a hierarchic structure. All the attributes, features and properties of selected elements are recorded in the elements list. The PBS is represented as an "AND" hierarchical tree, in which the overall relationships among parts and elements are indicated in this structure. Links in the structure are viewed as "a part of"(see Fig. 3).

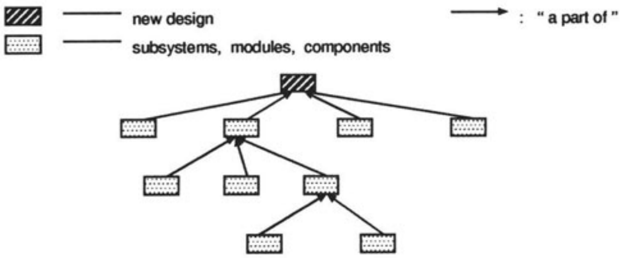

Figure 3 Product Breakdown Structure

In the above domestic heater case, the simplified heater breakdown structure is composed of *Motor* and *Control Unit* (see Fig. 4). In other words, *Motor* and *Control Unit* are "part of" *Heater.*

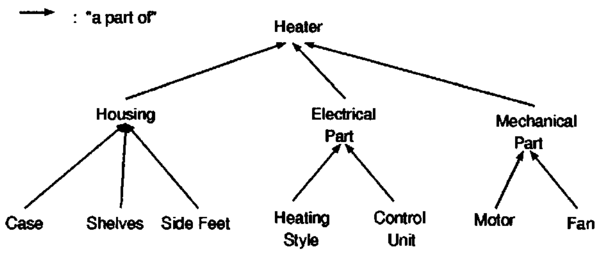

Figure 4 Domestic Heater Breakdown Structure

4 A REASON MAINTENANCE APPROACH

The purpose of the configuration support system is to aid a designer to manage the process of defining the configuration of new product. As an assistant, the approach of interactive decision support rather than automatic generation is adopted. This is in contrast to many systems for configuration design in which solutions are generated automatically based on a set of requirements.

The style of the system is interactive, allowing a designer to select components from a generic product structure, while maintaining the consistency of decisions to ensure the product will perform correctly.

4.1 Configuration Knowledge Within the System

The system specification is presented in terms of its input/output aspects based on modelling of configuration knowledge (see Fig. 5). In the configuration support system, PFCTs, design constraints knowledge and decision making sequence can be the inputs, whereas an instance of PBS and an explanation facility can be the outputs.

The product information can be structured as the PFCTs, and a set of constraints can be inputed as the constraint knowledge. The decisions form the interactive input during a design session. Each decision represents either a choice of a particular system, module or elements from the PFCTs, or a propositional logic combination in terms of assertions in logic terminology. A decision can be changed at a later stage in terms of the configuration process. After each decision the system propagates the effects of the decision using its PFCTs and constraints knowledge to maintain consistency. These inputs can be captured directly from the pre-defined knowledge base.

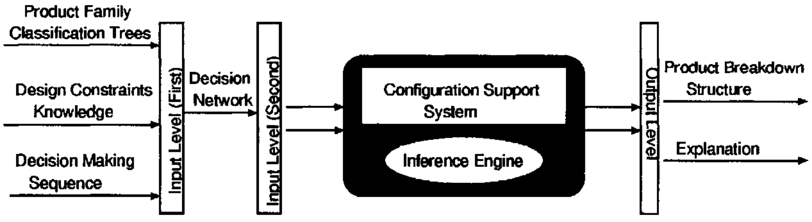
Product Family Classification Trees
Design Constraints Knowledge
Decision Making Sequence
Input Level (First)
Decision Network
Input Level (Second)
Configuration Support System
Inference Engine
Output Level
Product Breakdown Structure
Explanation

Figure 5 The Configuration Support System Input/output

There could be a methodology of classifying the PCFTs domain based on the defined structure. The product information can be structured into the proposed formalism based on different considerations. For example, from the product design point of view, the product information can be classified as the PCFTs structure based on functions or product performance. The product information can also be classified based on the customer requirements or application area from the user point of view.

As the configuration solutions, the product structure can be structured as the PBS formalism based on different aspects, such as functions and assembly etc. In a distributed manufacture for example, the product subsystems, modules and elements can be structured based on their manufacture locations.

4.2 The Configuration Design Process

In this formalisation of the system, configuration design is the process of deriving an instance of the PBS through a series of decisions on choice sets to best meet a set of requirements. The configuration design process therefore is shown to be the generation of an object which is composed of smaller sub-objects that together meet the required functionalities. The process starts from a set of requirements that need to be satisfied by the product to be configured, then produces a suitable decision network for the case which allows a designer to make decisions on elements choices selection. Eventually a product breakdown structure along with its explanations is produced based on the decisions.

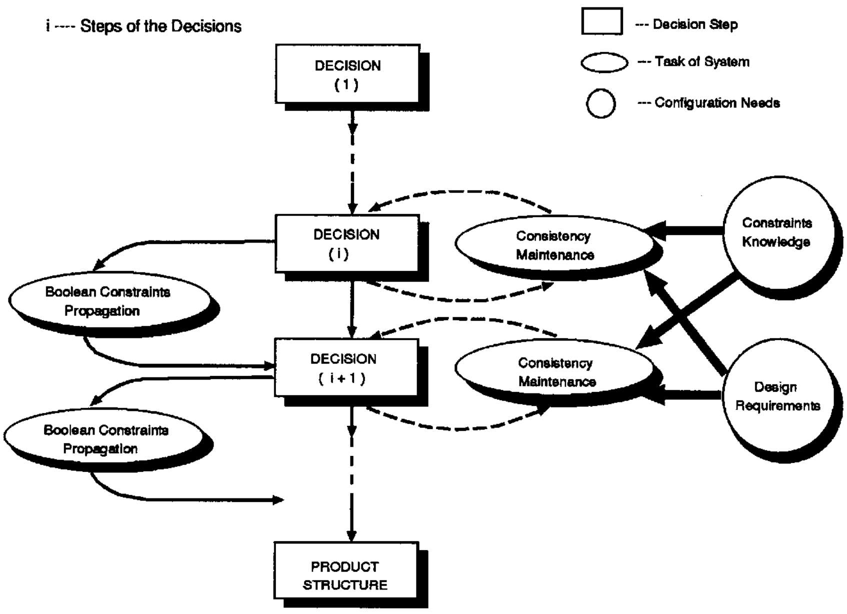

Figure 6 Configuration Procedure

Fig.6 illustrates the configuration procedure. During configuration design, each decision can be shown as an individual consistent statement that relates to the previous decision. Whenever a decision is made, the previous dynamic constraints will be tested to check either if they need to be relaxed or if new constraints are propagated. In addition, fixed, given constraints might be added depending on inference from previous decisions. On the other hand, the consistency among the decisions, and among the selected elements is maintained to ensure the validity of the decision sequence based on the constraints knowledge and design requirements. The process continues until the decision sequence satisfies all the requirements and constraints given by the designer.

The system searches through two entities of each nonleaf noed in the PFCTs to find suitable parts or elements. If there is no expected part in the parts breakdown entity, the system will ask the designer to define this new part for a new parts breakdown. For instance, normally *Storage Heater* is composed of two parts *Motor* and *Control Unit.* If the requirement says this heater will be a new model of storage heater with a fan system, the system will create a link between *Storage Heater* and *Fan System.* In this case, *Fan System* could be either in the PFTs or a new concept for the heater configuration.

As the solution of configuration, an instance of PBS is being created while the configuration process occurs. This product breakdown structure is an "AND" logical hierarchy, and combines all the selected components with "a part of".

4.3 Inference Engine Design

At the heart of the configuration management system, the inference engine design focuses on the logical dependencies handling within decisions and elements. Since consistency maintenance and constraints handling are the key problems in configuration design, the core of the inference engine is a truth maintenance approach in which a constraint-based reasoning technique assists in constraint management.

As the core of system, the *Inference Engine* controls the configuration process, makes inferences on the decisions and maintains the consistency among the components and decisions. It uses the Reason Maintenance System (RMS) which is a non-monotonic reasoning mechanism to assist the designer to do configuration. The purpose of the RMS is to assist the problem solver, which operates on a body of domain knowledge to make inferences according to some problem-solving procedure, making and maintaining these inferences(Dolye 1979)(McAllester 1980)(Kelleher 1988).

In order to realise this approach, the inference engine of the system is designed in three parts: *a Decision Processor*, *a Logic-based Truth Maintenance System* and *a Forward Checking algorithm* (see Fig.7). In this structure, the decision processor can be a problem solver which manipulates the configuration knowledge sources, guides the search, and creates the configuration solutions based on a series of decisions. The Logic-based Truth Maintenance system (LTMS) technique is used as the decision processor subsystem, to assist the decision processor to maintain consistency and dependency with change, and to handle logical constraints(McAllester 1980). The Forward Checking (FC) algorithm, as a constraint-based reasoning techniques, increases the power of the inference engine in constraint management(Kelleher 1988).

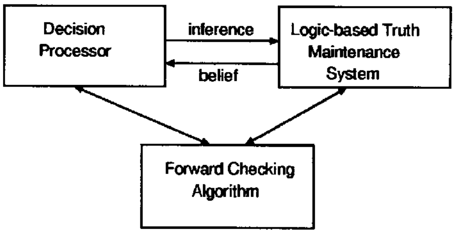
Decision Processor
inference
belief
Logic-based Truth Maintenance System
Forward Checking Algorithm

Figure 7 The Inference Engine Architecture

As a problem solver, the task of the decision processor is concerned with the manipulation of the configuration knowledge sources, as well as the maintenance of consistency and dependency either once a decision is made or a decision is changed. In addition, it supports the configuration task to obtain a solution, by checking if the selected components meet the requirements and constraints.

The LTMS maintains a table of decisions and justifications for the decisions. It deduces the new justification for a made decision in order to constrain the further unmade decision. Meanwhile, it checks for consistency of a decision, that is, revises the beliefs that cause contradiction in the made decisions. The Dependency Directed Backtracking algorithm is used to identify and retract the decisions that underlie a contradiction.

As a constraint-based reasoning technique, the Forward Checking algorithm is used to increase the power of the inference engine by constraint management. The algorithm eliminates incompatible branches early on in the search so that the efficiency is improved compared with chronological backtracking.

More details about the inference engine design has been described in (Yu 1994).

5 SUMMARY

A goal of the approach described in this paper is to develop a knowledge-based system prototype which will assist in the configuration management task. The product configuration knowledge modelling enables the product families, product breakdown structure and constraints knowledge to be structured as domain-independent formalisms. The system inference engine is able to make inferences relying on the product knowledge structure to identify parts or components which are selected to form a product structure. The idea of designing the inference engine is to use a RMS techniques which has a nonmonotonic reasoning mechanism to manage constraints and maintain consistency among selected components.

This approach has been tested on several examples of configuration domains, such as domestic heaters, telephone sets and sootblowers. In these examples, the product domain information has been structured into the configuration knowledge tree i.e. the *Product Family Classification Trees.* The inference engine accesses these trees along with a set of well-defined constraints to support the designer to make decisions on selecting elements for these product domains. It maintains consistency among elements and decisions during configuration, so it ensures that the breakdown structures of heater, telephone set and blower meet the design requirements and satisfy all the constraints. In practice, it has been shown that this approach is an appropriate mechanism for a configuration support system(Yu 1992)(Yu 1995). The whole configuration support system is being built by using this approach. The architecture which incorporates the reason maintenance approach and integrates with existing product structures is being developed for this purpose.

6 REFERENCES

Doyle, J. (1979) A Truth Maintenance System. *Artificial Intelligence,* No. 12, pp231-272,

Kelleher, G. and Smith, B.M. (1988) A Brief Introduction to Reason Maintenance Systems. *Reason Maintenance Systems and Their Applications,* Ellis Horwood Limited, ISBN 0-7458-0482-9, pp4-20,

de Kleer, J. (1986) An Assumption-based TMS. *Artificial Intelligence,* No. 28, pp127-162, de Kleer, J., Forbus, K. and Mcallester, D. (1989) Truth Maintenance system. Tutorial Notes from *IJCAI,*

Martins, J.P. (1991) The Truth, the Whole Truth and Nothing But the Truth. *AI Magazine,* Vol.11, NO.5, pp7-25,

MacCallum, K.J., Yu, B., Frederiksen, A. and McGregor, D. (1992) A System For Supporting Design Configuration. *Proceedings of Artificial Intelligence in Design'92,* Kluwer Academic Publishers, ISBN 0-7923-1799-8,

McAllester, D.A. (1980) An Outlook on Truth Maintenance. *AI Memo 551, MIT AI Laboratory,*

Pugh, S. (1991) Total Design. Addison-Wesley Publishing Company, ISBN 0-201-41639-5,

Yu, B. (1992) The Use of Artificial Intelligence Techniques for Configuration in Engineering Design. *Internal research report, CAD Centre, University of Strathclyde,* CADC/R/92-23, CM/R/92-03,

Yu, B. and MacCallum, K.J. (1994) Reason Maintenance for a Configuration Management System, *Expert System'94 Conference Proceedings: Research and Development in Expert Systems XI,* SGES Publications, pp173-185, ISBN 1-899621-01-6,

Yu B. and MacCallum, K.J. (1995) Decision Support for Configuration Management in Product Development, *The 3rd International Conference on Computer Integrated manufacturing (ICCIM'95),* Singapore, 11-14th July.

BIOGRAPHY

Bei Yu obtained her BSc degree of Computer Science in 1984. After one year training by Hitachi Japanese Company, she became a software engineer to carried out the industrial related research projects. Since 1991, she has been working as a Research Fellow in the CAD Centre, University of Strathclyde, and working on the research project of computer supported configuration design and management. She also is a PhD candidate and will be finishing off by 1995. Her main research interests are Artificial Intelligence Techniques, Knowledge-based systems, computer supported configuration design and management, product modelling and product structuring methodology.

Professor Ken MacCallum obtained his first degree in Naval Architecture from the University of Glasgow, proceeding to postgradute study in Imperial College, University of London where he obtained a PhD for research into the application of computer graphics to free-form surface design. After three years with a software company, he joined the University of Strathclyde, establishing the CAD Centre in 1985 as a research and postgraduate centre.

Ken MacCallum's main area of research has been the application of Artificial Intelligence to Engineering Design. He has led projects concerned with intelligent design modelling, data exchange, computer based design coordination, and computer aided learning. He is editor of the International Journal on Artificial Intelligence in Engineering, is a member of IFIP WG5.2, and has been on the Technical Programme Committees of a large number of Conferences and Workshops concerned with computer aided design. Ken MacCallum is currently the Head of Design, Manufacture and Engineering Management in the Faculty of Engineering at the University of Strathclyde.