國立交通大學

資訊科學與工程研究所

碩士論文

階層案例式電腦輔助製程規劃系統

A Hierarchical Case-Based Computer-Assisted Process-Planning System

研 究 生：徐啟珺

指導教授：曾憲雄 博士

中 華 民 國 九 十 八 年 六 月

階層案例式電腦輔助製程規劃系統

A Hierarchical Case-Based Computer-Assisted Process-Planning System

研 究 生：徐啟珺　　Student：Chi-Chun Hsu

指導教授：曾憲雄博士　　Advisor：Dr. Shian-Shyong Tseng

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2009

Hsinchu, Taiwan, Republic of China

階層案例式電腦輔助製程規劃系統

研究生: 徐啟珺　　　　　　　　指導教授: 曾憲雄博士

國立交通大學資訊學院

資訊科學與工程研究所

摘　要

由於龐大的客戶拓展以及快速的技術發展，近年來大量客製化在工業製造中扮演越來越重要的地位。然而，大量客製化時常伴隨著大量人力和時間成本的消耗。欲降低這些成本，應用電腦輔助製程規劃來降低製程規劃的複雜度不失為一種可行的方式。本篇論文著重於建立一套電腦輔助製程規劃系統。我們提出了一套使用階層案例式架構的系統，藉此幫助使用者管理工業製造領域知識，並提供產品設計師和系統之間的互動機制。在本篇論文中，我們提出了階層案例式電腦輔助製程規劃系統，並將之應用於手機製造產業。其中我們定義了階層案例的結構，並提出可以達到互動式案例調整的有效方式。此外，一個運用階層案例式電腦輔助規劃系統的案例，和其相關的實驗數據及討論等也會在這篇論文中加以論述。

關鍵字: 案例式推論、客製化、工業製造、製程規劃

A Hierarchical Case-Based Computer-Assisted Process-Planning System

Student: Chi-Chun Hsu **Advisor: Dr. Shian-Shyong Tseng**

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract

In recent years, mass customization becomes more and more important in manufacturing due to large amount of expanding customers and fast changing of developing techniques. However, mass customization often needs large amount of time and personnel costs. To reduce these costs, decreasing the complexity of process planning via applying a computer-assisted process-planning system is a feasible approach. This thesis focuses on the development of a computer-assisted process-planning system. This system utilizes hierarchical case-based techniques to facilitate managing various manufacturing domain knowledge and providing more interactions with designers. A hierarchical case-based computer-assisted process-planning system is then designed and implemented for mobile phone manufacturing in this thesis. The structure of hierarchical case structure is defined and the practical methods of iterative case adaptation process are presented. Illustrative examples and related discussion are also included to demonstrate the proposed system and its effects.

Keywords: Case based reasoning, customization, manufacturing, process planning

誌　謝

這篇論文的完成，必須感謝許多協助與支持我的人。首先要感謝我的指導教授，曾憲雄博士，在我碩士班這二年的時間當中，由於老師極具耐心的指導與勉勵，使我在研究方法、思考方法以及表達方法等等各方面獲益良多，不甚感激！同時也要感謝我的口試委員，楊鎮華教授、洪宗貝教授以及袁賢銘教授，對於此篇論文提出許多寶貴的意見，讓此篇論文能更臻完善。另外也要感謝工業研究院谷中昭先生、蔡瑞彬先生以及亞洲大學教授張峯銘教授在本篇論文撰寫期間提供大量的專業知識和文獻資料，使本篇論文更加豐富、更加有價值。

接下來要感謝蘇俊銘學長、翁瑞鋒學長、林喚宇學長不厭其煩的指出我研究中的缺失，並在我迷惘困惑時不辭辛苦的為我解惑。在學長們的指導與幫助之下，此篇論文才能漸漸成型、完成，並讓我在過程中體驗做研究的樂趣以及學習做研究應該具有的態度。同時也感謝其他學長，尤其是李威勳學長、董元昕學長、司馬伊凡學長，在本篇論文修定期間，對本篇論文中的論述提出許多精闢的見解以及寶貴的意見。也感謝惠君、靖雅、祖淵、士緯、世恆等同學的幫忙及互相勉勵，陪伴我度過充實的碩士生涯。並同時感謝嘉祥及佳榕等學弟妹，在我忙碌的時候熱心的幫忙分擔一些事務，使我負擔減輕許多。

另外要感謝我的家人們在背後默默地支持我的學業，讓我可以安心的做研究無需擔心生活上的問題。最後要特別感謝我的女朋友佳樺，一直在我身邊不時地關心我、照顧我、鼓勵我，讓我能夠自信、樂觀的面對一切難題。當然也非常感謝這兩年間在我身邊和我一起奮鬥成長的朋友們，由於人數眾多不及備載，在此我向你們說聲，謝謝！

Table of Content

摘　　要......i

Abstract......ii

誌　　謝......iii

List of Figures......v

List of Tables......vi

Chapter 1. Introduction......1

Chapter 2. Literature Review......4

- 2.1 Approaches to process planning......4
- 2.2 Previous work......5

Chapter 3. HCB-CAPP System......7

- 3.1 Product Process Ontology......7
- 3.2 HCB-CAPP System Architecture......8
- 3.3 Hierarchical Case Representation......9
- 3.4 Feature Concept Hierarchy......13

Chapter 4. Planning Algorithm of HCB-CAPP System......14

- 4.1 Hierarchical Case-Based Process Planning Algorithm......14
- 4.2 Intelligent Query Processor......15
- 4.3 Similarity of Cases......19

Chapter 5. Collaborative Environment......21

Chapter 6. System Implementation and Experiment......23

- 6.1 System Implementation......23
- 6.2 Experiment Design......26
- 6.3 Experiment Result and Discussion......27

Chapter 7. Conclusion......32

References......33

List of Figures

Figure 1: Product Process Ontology for Mobile Phone Manufacturing 7

Figure 2: The scenario of mobile phone manufacturing product process design 8

Figure 3: HCB-CAPP system architecture...9

Figure 4: Case tree of product IRIT2009 12

Figure 5: A partial feature concept hierarchy to describe the feature *Color* 13

Figure 6: The process of finding appropriate case tree (1/4) 17

Figure 7: The process of finding appropriate case tree (2/4) 17

Figure 8: The process of finding appropriate case tree (3/4) 18

Figure 9: The process of finding appropriate case tree (4/4) 18

Figure 10: Feature similarity of *Dark Red* and *Dark Yellow* 20

Figure 11: The Questionnaire Results of Satisfaction Degree 29

Figure 12: The Questionnaire Results of Satisfaction Degree 29

List of Tables

Table 1: The list of data type and possible values in the six feature slots 10

Table 2: The Format of a Five-Level Linkert Item 27

Table 3: Question Items in Questionnaire for Satisfaction Degree..............................28

Table 4: Question Items in Questionnaire for Advancement Degree............................28

Chapter 1. Introduction

Recently, in the modern globalized environment, the traditional strategy of manufacturing industries, called Economies of Scale, is not suitable to the highly changeable market, such as the market of mobile phone. Because of the short product life cycle of modern mobile phones, the competitiveness of enterprises depends on the ability to flexibly and rapidly response to the customers' new requirements. How to efficiently plan the manufacturing processes and machines to meet the customization becomes an important challenge of enterprises [10]. To design manufacturing process needs to determine the features of product, design the process to satisfy the requirements of product, and schedule the machines to complete the used process. Since it is difficult for designers to integrate these three kinds of expertise to plan a new manufacturing process, how to manage the multi-domain expertise and assist designers to integrate these kinds of knowledge for a new product design is an important issue. Case-based approach is an appropriate mechanism to reuse and adapt the domain expertise for new requirements, so we aim to apply case-based approach to reuse the previous manufacturing process design knowledge and assist designers to integrate the retained knowledge for new product manufacturing. According to our observation, when designing a new product, most of the designers use the rapid prototyping approach to find the similar products at first and then adapt the corresponding process and machines to satisfy the detailed requirements. Thus, the three kinds of domain knowledge are suitably represented as a multi-level case structure, which represents a case in product level, process level, and machine level, so designers can retrieve the case according to the features in product level, and adapt the case in the lower levels of process and machine, sequentially.

The previous researches [3][4] aim to apply case-based mechanism to plan manufacturing process, but the flat case structures are difficult to manage the multi-domain expertise. Besides, in these researches, the case-based systems are a black box, so designers need to determine the requirements from product level to machine level before case retrieval and cannot interact with case base to refine the requirements and results in the case adaptation process. [13] proposed a Case-based process planning system, using divide-and-conquer approaches to solve the problems of multiple domains, but the black box case adaptation process still cannot provide interaction with designers. [11][2][12][8] applies hierarchical case-based approaches to manage the multi-level cases, but these researches focus on how to use hierarchical structure to enhance the retrieval performance and lack a mechanism which can interact with designers in case retrieval and adaptation to allow users to refine the retrieved requirements and results in different knowledge levels.

There are two issues raised in the previous case-based manufacturing process design systems:

- How to manage multi-level domain expertise.
- How to interact with designers to facilitate them to determine the detailed design in the case adaptation process.

Accordingly, a case-based process planning systems needs multi-level case structure to facilitate managing various manufacturing domain knowledge. When a new product needs to be designed, the system should interact with designers to adapt the previous cases from coarse-grained to fine-grained. Therefore, we propose a **Hierarchical Case-Based Computer-Assisted Process Planning System (HCB-CAPP)**, where the hierarchical case structure is used to model the multi-level manufacturing knowledge and an iterative case adaptation process is provided for designers to input detailed requirements and modify the retrieved cases from product level, process level, to machine level. In HCB-CAPP system, the product manufacturing design cases are modeled as **Product Process Ontology (PPO)** which contains the cases of product level, process level, and machine level, and the relations between these cases. Based on product process ontology, an iterative case adaptation process is conducted by an **Intelligent Query Processor (IQP)**, which can retrieve appropriate cases to meet the designers' requirements from coarse-grained to fine-grained, and interact with designers to modify requirements and retrieved cases in each level. In the case retrieval, besides the primitive feature types, such as numerical, enumerative, and string-based feature, we also define the ontology-based features, whose similarity can be defined in the feature ontology, so the features' correlations can be managed by domain experts.

Finally, we construct an HCB-CAPP system for mobile phone manufacturing, and the system introduction and discussion will be provided in this thesis. We also invite some Mobile Phone Product Designers, Process Planner, and Machine Manager to make use of our designed system and to offer some useful advice and comments. Then, we design questionnaire based on some software evaluation metrics and request some mobile phone manufacturing domain experts and some programmers to evaluate our system performance.

The remainder of the article is organized as follows. In Chapter 2, we introduce some related works about the approaches to process planning. Then, the proposed HCB-CAPP system scheme and planning algorithm are described in Chapter 3 and Chapter 4 respectively. Chapter 5 introduces the collaborative environment system construction to facilitate knowledge management. The implementation for HCB-CAPP and experiments are discussed in Chapter 6. Finally, Chapter 7 gives the conclusion and future work.

Chapter 2. Literature Review

2.1 Approaches to process planning

Generative process-planning systems automatically synthesize a process plan for a new component. They generate a process plan for a part from scratch, based on manufacturing information stored in a database, and decision-making logic and algorithms. However, the generative approach does not utilize the experience gained from past solutions. Also, they generally lack the option of generating alternate plans for a given part, and system knowledge cannot be extended or modified without significant reprogramming. A case-based process planner avoids the duplication of solution effort found in generative system by reusing past experiences to solve new problems. Old solutions are retrieved and adapted to fit the new scenario.

Variant process-planning systems group parts into a family. Plans for a family of parts are stored in files. A new part is classified into one of the families. A standard plan for that family is retrieved and manually modified, based on the new part's dimensions and features. However, variant approaches are labor intensive to implement. They work only for parts similar to those planned previously. Experienced process planners are still required to modify the standard plan; hence, variant process planners are not completely automated. A case-based system reuses previous experiences automatically to generate new solutions.

2.2 Previous work

Most previous approaches to computer-aided process planning (CAPP) can be categorized into either variant methods [9] or generative methods [18]. Because the vast number of works published in this area, we briefly review a small subset, which are mainly knowledge-based approaches. GARI [20] is a metal-cutting constraint-based planner that creates a loosely constrained plan and then iteratively constrains it, using expert knowledge and backtrackings. SIPS [6], which is developed in LISP, uses a best-first branch-and-bound strategy to find the best-cost plan. EXCAP [1], developed in England, is a rule-based system which plans for rotational components utilizing backward chaining. Hummel [16] proposed the design of XCUT, which was developed in the Allied Signal Corporation. XCUT generates an object-oriented description of feature using feature taxonomy. It is coupled with a production system for extracting relevant information from feature volume representation. Other knowledge-based systems include IMACS [7], FBAPP [14], IOOPP [15], and Joshi et al. All of these systems belong to the category of generative approaches. Group technology is the most popular effort towards the development of CAP systems based on the variant approach.

The first attempt to develop a case-based approach to process planning was reported by Tsatsoulis and Kashyap [3]. They describe a system for rotational parts which consider the machining surface one at a time. Other case-based process planners for rotational parts include Yang et al., Humm et al. and Bergmann and Wilke. Zarley developed a case-based process planner for assembly operations, and et al. investigated case-based process planning in metal forming. Tiwari et al. [17] proposed a case-based CAPP system for machining prismatic components. Felix uses an approach that combined with not only RBR, but also FL concept to generate better result on electroplating industry.

Researches above attempts to apply a flat case-based structure, it might have problems for a well-structure knowledge management that includes multi-level hierarchical characteristics. According to our observation, some domain knowledge contains hierarchical characteristics. Using flat case-based structure is nontrivial for management this kind of domain knowledge. Moreover, designers' queries were difficult to be designed, because they need to know the details of the whole production process when providing requirements. Besides, the black box approaches are difficult to allow designers to modify the result in adaptation process. Therefore, we propose a hierarchical case base planning approach to facilitate reusing previous design cases from coarse-grained to fine-grained base upon Product Process Ontology and similarity function.

Besides applying hierarchical case-based, our work is essentially different from those mentioned above systems because it is developed for Mobile Phone Manufacturing. In Mobile Phone Manufacturing, we can easily found the apparent hierarchical characteristics of the knowledge.

Chapter 3. HCB-CAPP System

In this thesis, we aim to propose a Hierarchical Case-Based Computer-Assisted Process Planning system (HCB-CAPP) which can manage multi-level cases and allow users to provide requirements and modify retrieved cases from coarse-grained to fine-grained.

3.1 Product Process Ontology

After interviewing with mobile phone product designers, we define a three-layer product process ontology to represent the domain knowledge for Mobile Phone Manufacturing:

- Product layer represents the products' relationship.
- Process layer represents the process information for some specific product.
- Machine layer represents the information about machines used by a specific process.

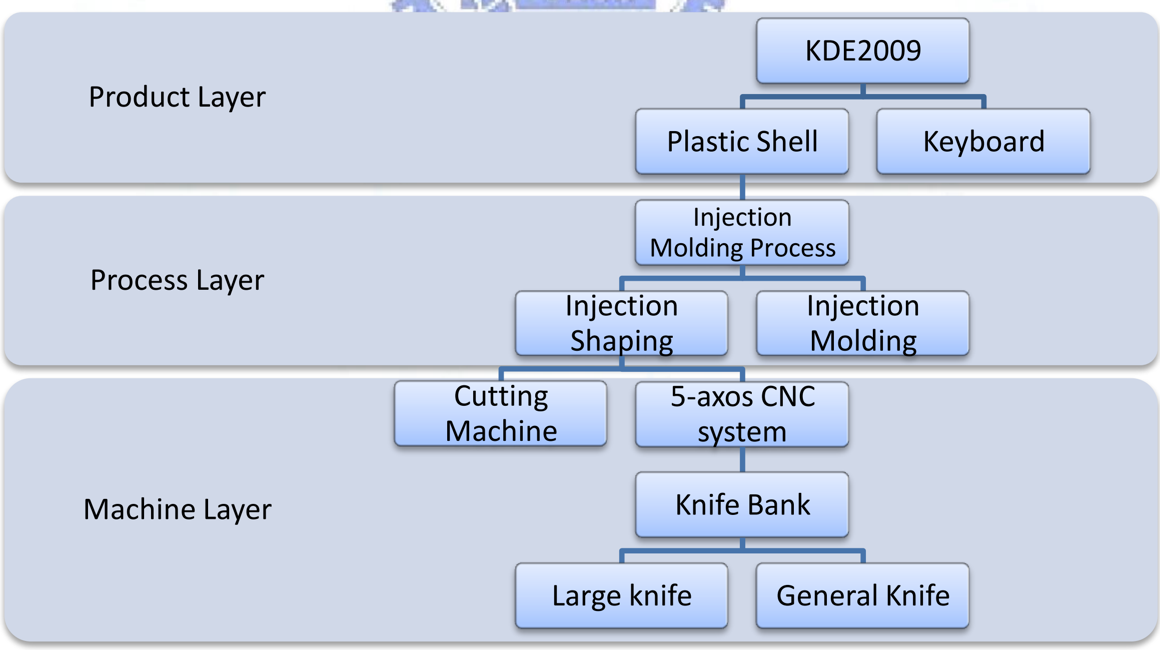


Figure 1: Product Process Ontology for Mobile Phone Manufacturing

3.2 HCB-CAPP System Architecture

As described above, product designer will provide new requirements and then search for the most similar previous case to reuse. If the most similar case can't fulfill new requirements, product designer may revise the result instead of directly reusing. Finally, the case will be retained for future reuse, as shown in Figure 2.

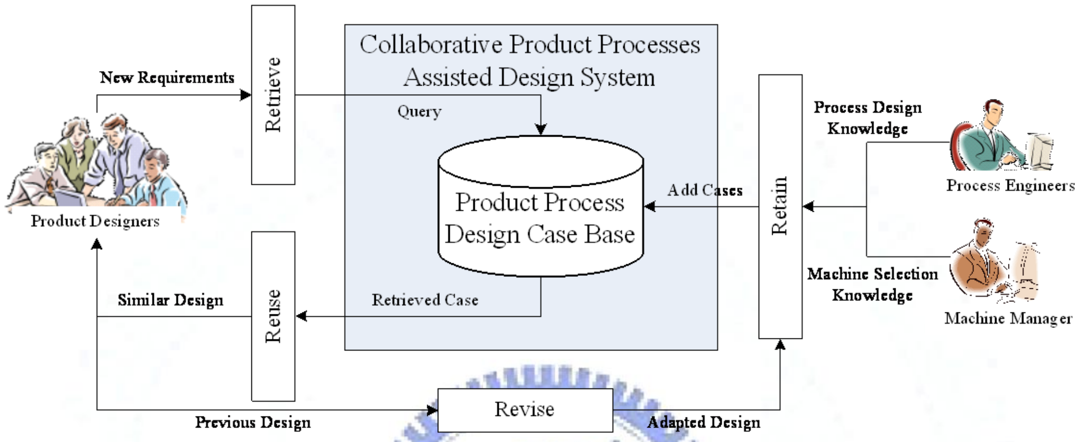


Figure 2: The scenario of mobile phone manufacturing product process design

In order to facilitate not only the retaining and integration of product, process, and machine domain knowledge in the case base but also requirements providing and result modifying by designers in a lower complexity way, we construct a collaborative product process assisted design system.

In our Hierarchical Case-Based Computer-Assisted Process-Planning (HCB-CAPP) system, as shown in Figure 3, the process plan can be retrieved from coarse-grained to fine-grained for higher relevance to the product requirements.

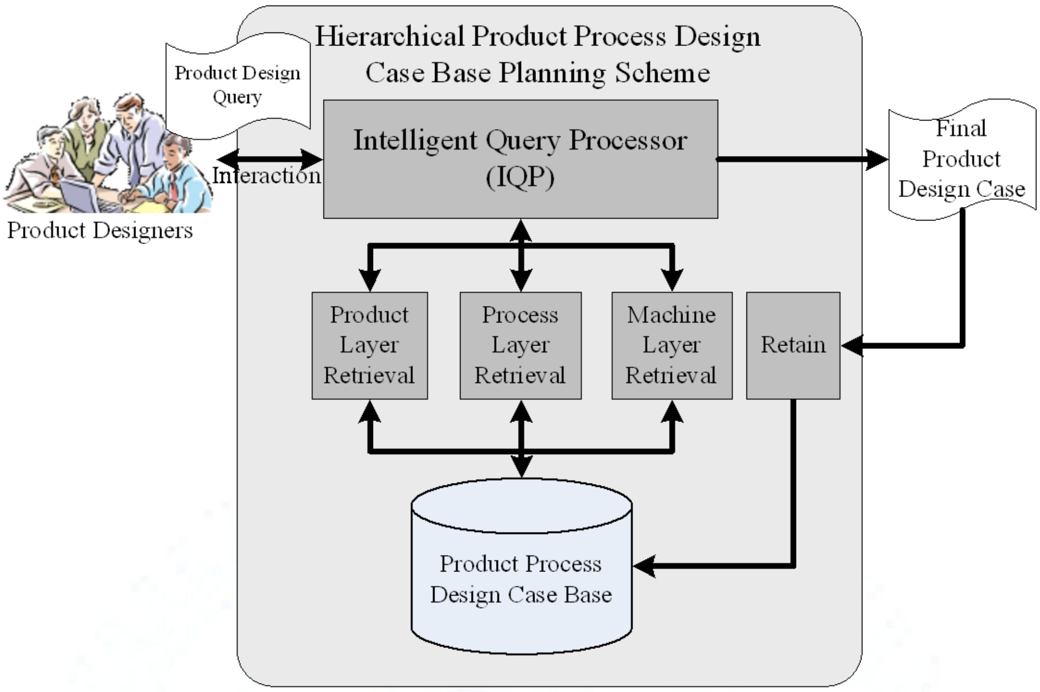


Figure 3: HCB-CAPP system architecture

3.3 Hierarchical Case Representation

As mentioned above, the process planning in mobile phone manufacturing can be represented in three granularity levels: product level, process level, and machine level.

The definitions of feature vectors are shown in Definition 1, where FV_i is the feature vector of i_{th} case node, and v_{ij} is the feature value of feature f_j, v_{ij} can be a possible values V_j, or null if the feature f_j is not required to describe the case node.

Definition 1: Feature Vector (FV)

- Let $f_1, f_2, \ldots, f_n$ describe all the case nodes.
- Let V_j be the set of all possible value of f_j
- $FV_i = \langle v_{i_1}, v_{i_2}, \ldots, v_{i_n} \rangle$, for $v_{i_j} \in V_j \cup \{null\}$

Since different subsets of features may describe different case node, the feature function $\Phi(FV_i)$ is defined to denote the set of non-null features f_j in FV_i

Definition 2: Feature Function (Φ)

- $\Phi(FV_i) = \{f_i | v_{i_j} \neq null\}$

Definition 3: Feature Vector Subsequence Relation (∝)

- $FV_i \propto FV_j \; iff \; \Phi(FV_i) \subseteq \Phi(FV_j)$

Example 1. The features of a mobile phone product.

To simplify the discussion, in the rest of this thesis, we assume there are six features describing the mobile phone product, where f_1=“Name”, f_2=“Describe”, f_3=“Purpose”, f_4=“Material”, f_5=“Component”, f_6=“Size”. The data type and possible values of each slot are shown in Table 1.

Table 1: The list of data type and possible values in the six feature slots

Slot	Data type	Possible values
Name	string	{“IRIT2009”, “KDE2009”}
Describe	string	{“Developed by IRIT lab in 2009”, “Developed by KDE lab in 2009”}
Purpose	string	{“Alpha”, “Close-Beta”, “Open-Beta”, “Stable”}
Color	string	{“Dark Red”, “Light Red”}
Component	string	{“LCD”, “TFT-LCD”, “Keyboard”, “Camera”, “Shell”, “Battery”}
Thickness	number	{16mm,1 4mm}

Here we define the Case Tree to simplify our following discussion and make the result of our algorithm become clearer. A case tree can be divided into two set: Nodes N and Relations R. A case node n_j contains its own case node type $CNtype$ and a feature vector FV.

Definition 4: Case Tree (*CT*)

- Based on the three levels of product process ontology, $N = \{n_1, n_2, \dots, n_n\}$ denotes the set of the union of products, processes, and machines.
- $n_j = (CNtype_j, FV_j)$, for *CNtype* is the case node type and $CNtype_j \in \{product\,|process|machine\}$
- $R = \{< \mathrm{n_i}, \mathrm{n_j} >, < \mathrm{n_k}, \mathrm{n_l} >, \dots,\}$ denotes the relations between the case node pairs.
- $CT = (N, R)$ denotes the case tree.
- The type of a case tree is the same as the *CNtype* of the root of this case tree.

Example 2: The case tree of a mobile phone product.

As shown in Figure 4, the case tree can be generated from Product Process Ontology (see Figure 1) via a straightforward way. Here we can see the case tree contains a case node with CNtype = “product”, named “IRTI2009”, two case nodes with CNtype=”process”, named “Injection Molding Process” and “Assemble Process” respectively, and three case nodes with CNtype=”machine” belongs to them.

Each case node has its own case node type CNtype and feature vector FV respectively. And the relation set R is equal to {<IRIT2009, Injection Molding Process>, <IRIT2009, Assemble Process>, <Injection Molding Process, Injection Machine>, <Injection Molding Process, Cutting Machine>, <Assemble Process, Robot>}.

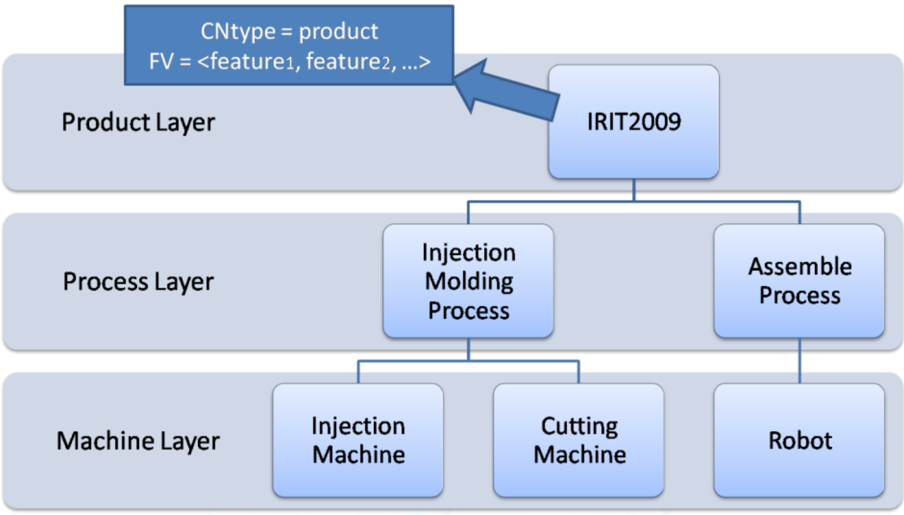


Figure 4: Case tree of product IRIT2009

To retrieve the required case tree, designer can define required case tree root node type, $CNtype_{req}$, and use the query feature vector (QFV) to describe their required features in the case tree.

Definition 5: Query Feature Vector

- $QFV_i = \langle v_{i_1}, v_{i_2}, \ldots, v_{i_n} \rangle$, for $v_{i_j} \in V_j$, and $\Phi(QFV_i) = \Phi(FV_k)$, if $CNtype_{req} = CNtype_j$

Example 3: The query feature vector to describe a mobile phone product.

In this example, the following query feature vector is defined to describe a required new product, which name is "KDE2010", describe is "Developed by KDE lab in 2010", purpose is "Close-Beta", component is {"LCD", "Keyboard", "Shell", "Battery"} and size is 50mm.

QFV = <"KDE2010", "Developed by KDE lab in 2010", "Close-Beat", {"LCD", "Keyboard", "Shell", "Battery"}, 50mm>

3.4 Feature Concept Hierarchy

By our observation, some features imply hierarchical characteristics. For these features, model them with a feature concept hierarchy can make the similarity computation more reasonable. We aim to organize the feature values in all possible feature values set V as a feature concept hierarchy, and use the feature values' distance in the concept hierarchy to determine these feature values' similarity. Figure 5 shows a partial feature concept hierarchy to describe the values of feature *Color*.

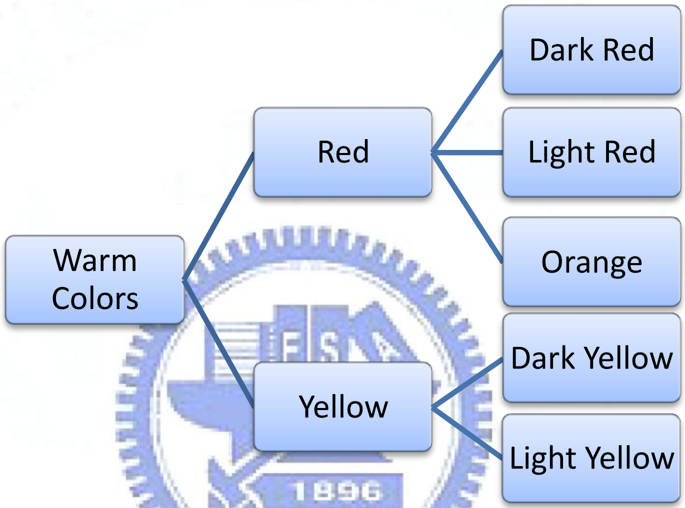


Figure 5: A partial feature concept hierarchy to describe the feature *Color*

By using feature concept hierarchy, we can organize and manage features more easily. And also, we can determine the relation or similarity between features more easily. We will discuss about how to determine the similarity between features in Chapter 4.

Chapter 4. Planning Algorithm of HCB-CAPP System

4.1 Hierarchical Case-Based Process Planning Algorithm

As shown in Algorithm 1, the inputs include the query feature vector (QFV), the required case tree type (CNtype), such as product, process, or machine. Afterward the Intelligent Query Processor can retrieve and adapt the final synthesized case tree in different levels according to the query feature vector and required case tree type. After designer reviews and revises the final synthesized case to fulfill the conditions of the current environment and added to the case base. Thus the HCBPP can cope with more complex or difficult requirements by incrementally increasing the number of cases in the case base.

Algorithm1 : Hierarchical Case-Based Process Planning (HCBPP)

Definition of Symbols:

QFV: The query feature vector, inputted by designers.

CNtype: The required case tree type, such as product.

Input: QFV, CType

Output: the final synthesized case tree

Step 1: Receive the input query feature vector and query case tree root type.

Step 2: Call IQP (QFV, CNtype) to get the synthesized case tree.

Step 3: Designers revise the synthesized case to fulfill the current enviroment.

Step 4: Retain the final synthesized case to the case base.

The most important process, IQP, which can retrieve different hierarchical cases and synthesize them to construct final synthesized cases in the Step 2 of HCBPP, will be discussed as follows.

4.2 Intelligent Query Processor

As shown in Algorithm 2, case node similarity function (CNSIM) is proposed to evaluate the similarity between the inputted query feature vector (QFV) and the retrieved case node. Then the system provides a sorted list of case trees made by these case nodes as its root respectively. Designer can choose the appropriate case tree and then revise this chosen case tree from top level to down level (ex. Apply level order traversal). If any inappropriate sub-case tree is found, apply intelligent query processor algorithm recursively to find an appropriate sub-case tree to replace the inappropriate one. It recursively executes above steps until no inappropriate sub-case tree can be found. At the end, store the case tree and finish the intelligent query processing. An illustrative example is shown in Example 4.

Algorithm2 : Intelligent Query Processor (IQP)

Definition of Symbols:

QFV: The query feature vector, inputted by designers.

CNtype: The required case type, such as product, process, or machine.

SCTList: A list of similar case trees, and the initial value of SCTList = ∅ .

Input: QFV, CNtype

Output: the final synthesized case tree

Step 1: $\forall c_i \in C, CNtype_i = CNtype$ insert c_i into SCTList by descending order of $CNSIM(FV_x, QFV_y)$

Step 2: Designer chooses the most appropriate case tree from SCTList, we denoted it as MACT

Step 3: Designer revises MACT from top level to bottom level, if there is a sub-case tree that is not appropriate for the requirements. We denoted it as NASCT. If there is no such a sub-case tree, go to Step 7.

Step 4: Designer can modify NASCT as NASCT', if NASCT' is appropriate for the requirements, go to Step 3..

Step 5: If NASCT' is still not appropriate for the requirements, designer can modify or add more details to the QFV as QFV'.

Step 6: Replace the NASCT by the output of IQP (QFV', ctype of NASCT'), then go to Step 3.

Step 7: Store the case tree.

Step 4: Retain the final synthesized case to the case base.

Step 4: Retain the final synthesized case to the case base.

Example 4: Use Intelligent Query Processor to find an appropriate case tree

First of all, a product designer will provide their requirements *R*, IQP take *R* and form a virtual case node *C* that contains all requirements *R*. Then IQP starts to calculate the similarity between *C* and the case trees CT_i in the case base. After this stage, a list of similar case trees' list *SCTList* will be generated. Designer now can choose the most appropriate case tree. In this example, we assume here the designer picks up the case tree rooted by IRIT2009. This process is shown in Figure 6.

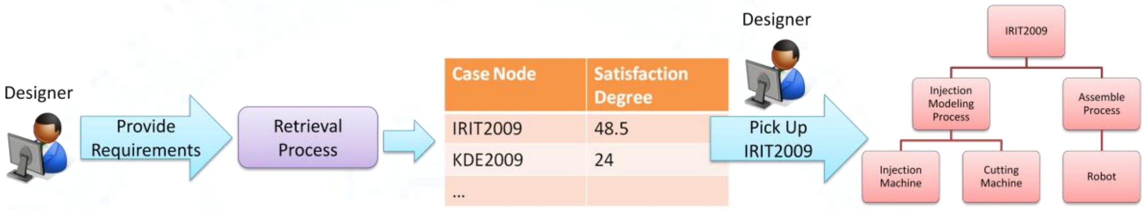


Figure 6: The process of finding appropriate case tree (1/4)

Secondly, designer can begin to revise the retrieval case tree level by level. Designer might evaulate each node and its attributes in indetail. If there is one node unsuitable for the product design, we mark it as a target that will be manipulated in next iteration. This process is shown in Figure 7.

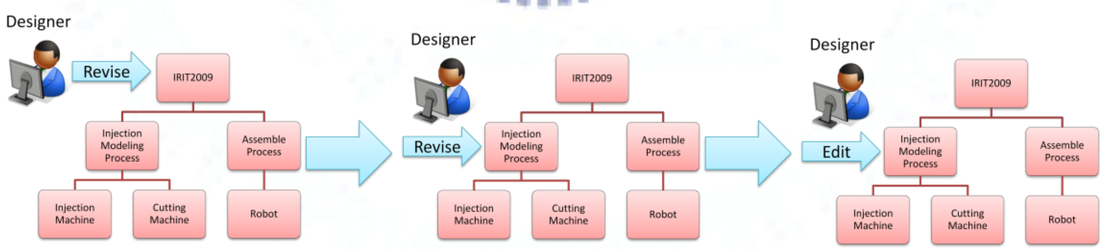


Figure 7: The process of finding appropriate case tree (2/4)

After we mark the target unsuitable node, requirements will be edited and then start another iteration of case retrieval. And also, the current case tree will also be included in the requirements. With new version of requirements, we apply the case retrieval process again and then find an appropriate sub-case tree. This process is shown in Figure 8.

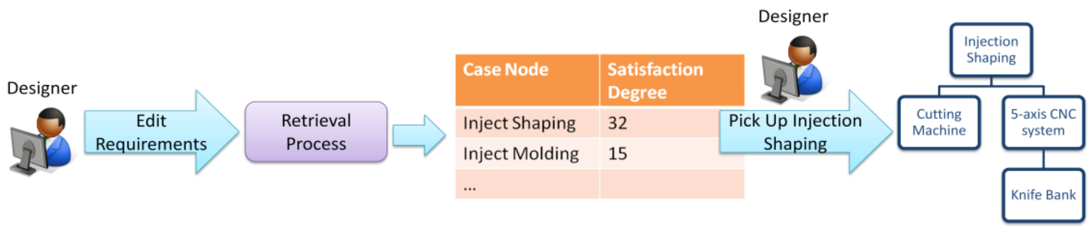


Figure 8: The process of finding appropriate case tree (3/4)

After previous process, now we can directly replace the original unsuitable case node with the root of new retrieval case tree. This process is shown in Figure 9.

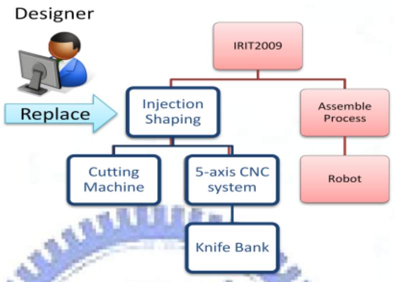


Figure 9: The process of finding appropriate case tree (4/4)

Repeating above process iteratively, finally we will form a case tree that can fulfill our requirements. In following section, we discuss about how to compute the similarity in retrieval phase.

4.3 Similarity of Cases

Here we apply a hybrid similarity function that includes Rule-Based, Ontology-based, Enumerated, Numeric and Symbolic feature similarity computing to calculate the similarity between cases.

Based on different feature types, we defined several different feature similarity functions respectively for different feature types' physical meaning. As shown below, we define four formulas, are defined to calculate the similarity of two feature values according to its type respectively.

Formula 1: Real Number Feature Similarity (RNFSIM)

$$\mathrm{RNFSIM}(v_i^I, v_i^R) = \frac{1}{1 + |v_i^I - v_i^R|}$$

where $v_i^I \in FV_I$, $v_i^R \in FV_R$.

Formula 2: String Feature Similarity (SFSIM)

$$SFSIM(v_i^I, v_i^R) = \frac{1}{1 + Levenshtein(v_i^I, v_i^R)}$$

where $v_i^I \in FV_I$, $v_i^R \in FV_R$, and the function *Levenshtein()* is use to calculate the minimum number of operations needed to transform v_i^I into v_i^R.

Formula 3: Ontology-Based Feature Similarity (OBFSIM)

$$OBFSIM(v_i^I, v_i^R) = \frac{1}{1 + Dist(v_i^I, v_i^R)}$$

where $v_i^I \in FV_I$, $v_i^R \in FV_R$ and the function Dist() is used to calculate the shortest distance between v_i^I, v_i^R and their common ancestor concept.

Formula 4: Enumerated Feature Similarity (EFSIM)

$$EFSIM(v_i^I, v_i^R) = cosine(v_i^I, v_i^R) = \frac{v_i^I \cdot v_i^R}{|v_i^I||v_i^R|}$$

where $v_i^I \in FV_I$, $v_i^R \in FV_R$.

Base on the feature similarity, the similarity between the query feature vector and the feature vector in the case node. Formula 5 shows the formal definition of Case Node Similarity Function.

Formula 5: Case Node Similarity Function (CNSIM)

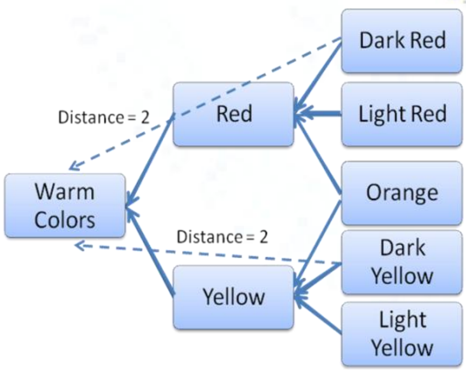
$$CNSIM(FV_x, QFV_y) = \frac{\sum_{i=1}^{n} FSIM(v_i^I, v_i^R)}{|FV_x|}$$

where $FSIM \in \{RNFSIM \mid SFSIM \mid OBFSIM \mid EFSIM\}$, $FV_x \propto QFV_y$

Example 5: Case unit similarity calculation

This example shows the ontology-based feature similarity (OBFSIM) calculation of two values v_i^I, v_i^R of feature f_k, where $f_k = \text{"Color"}$, $v_i^I = \text{"Dark Red"}$, $v_i^R = \text{"Dark Yellow"}$, based on feature concept hierarchy. In Figure 6, the common ancestor of v_i^I and v_i^R is "$Warm\ Color$", and $Dist(v_i^I, v_i^R)$ is the shortest distance between the shared ancestor and v_i^I, v_i^R, so $Dist(v_i^I, v_i^R) = 2$. Thus the feature similarity of "$Dark\ Red$" and "$Dark\ Yellow$" can be calculated as 0.333 by $OBFSIM(v_i^I, v_i^R)$.

$v_i^R = \text{"Dark Yellow"}$



$v_i^I = \text{"Dark Red"}$

$v_i^R = \text{"Dark Yellow"}$

$$OBFSIM(v_i^I, v_i^R) = \frac{1}{1 + Dist(v_i^I, v_i^R)}$$

$$= \frac{1}{1+2} = 0.333$$

Figure 10: Feature similarity of *Dark Red* and *Dark Yellow*

Chapter 5. Collaborative Environment

Performance of a case-based process planning system often depends on the quality and quantity of cases in the case base and our proposed system does, either. In order to obtain better cases, collaborative environment is a feasible solution since this way can involve more experts to offer their expertise and opinions and therefore cannot only gather more cases but also guarantee the quantity.

Wiki-like system scheme is suitable for construction of collaborative environment and also provides some benefits to facilitate knowledge maintenance. The main advantage of Wiki is the ability to allow users work in one space, upload documents and images and reference link to other topics for further reading. Our wiki-like system scheme also keeps these advantages and makes some changes to fit our HCB-CAPP system. Besides, we also try to facilitate version control and conflict handling

Version control is a very important mechanism in our wiki-like system scheme. Because materials in our wiki-like can be accessed by different users, version control mechanism offers a safe, recoverable, traceable function to prevent content corruption.

In our wiki-like system scheme, the version control mechanism provides following functions:

- Atomic operation
- Revision history
- Source browsing
- Source indexing
- Reports
- Access control

With the mechanism of version control, we can prevent content corruption. However, if some users have different opinions about one material, it might be possible to occur infinite editing for the same material. In this situation, we offer an extra page belongs to this material, named “Talk.” Users with different opinions can discuss in this extra page to find a way that they all can accept and then apply this common opinion to that material.

The conflict handling mechanism described above provides a way for users to communicate with others without disturbing the original material and to find a common opinion in an intuitive way.

Chapter 6. System Implementation and Experiment

6.1 System Implementation

With the product process ontology and feature concept hierarchy defined for Mobile Phone Manufacturing, we have constructed an experimental system of HCBPP. The interfaces of the HCBPP system are shown in Figure 7. And the steps of the HBPP system to generate an appropriate case tree are shown in Figure 8.

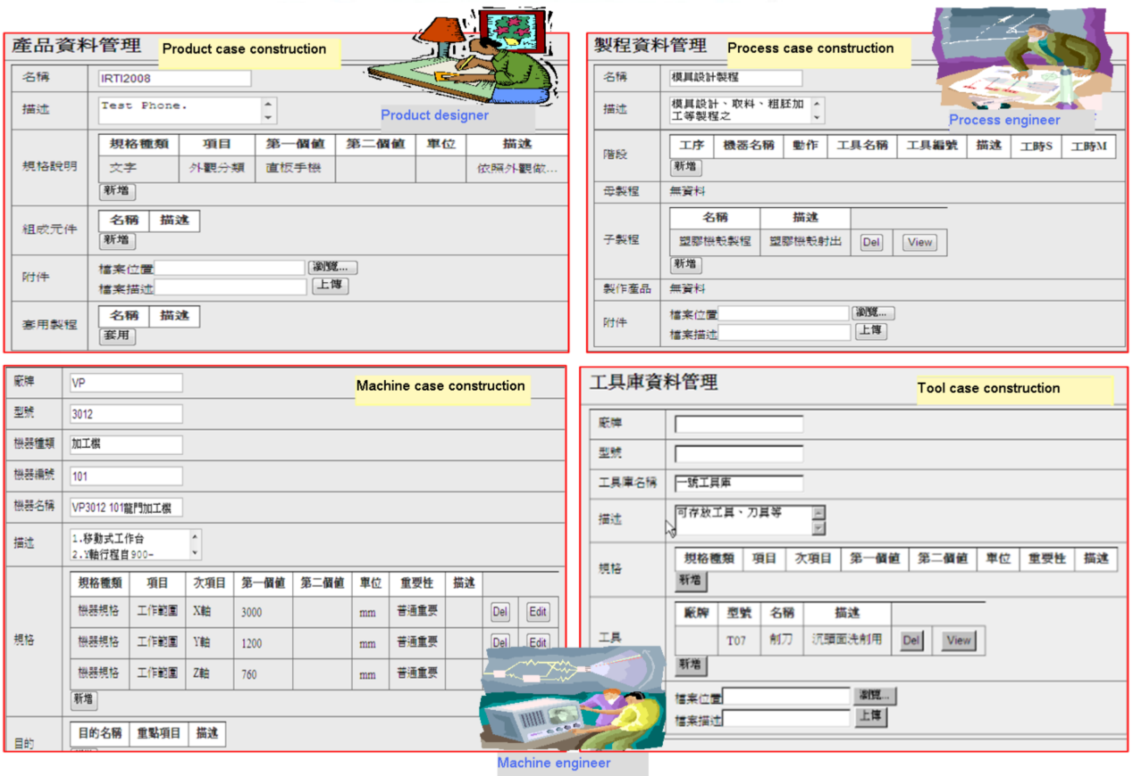


Figure 7: The input interfaces of query feature vector for different case type

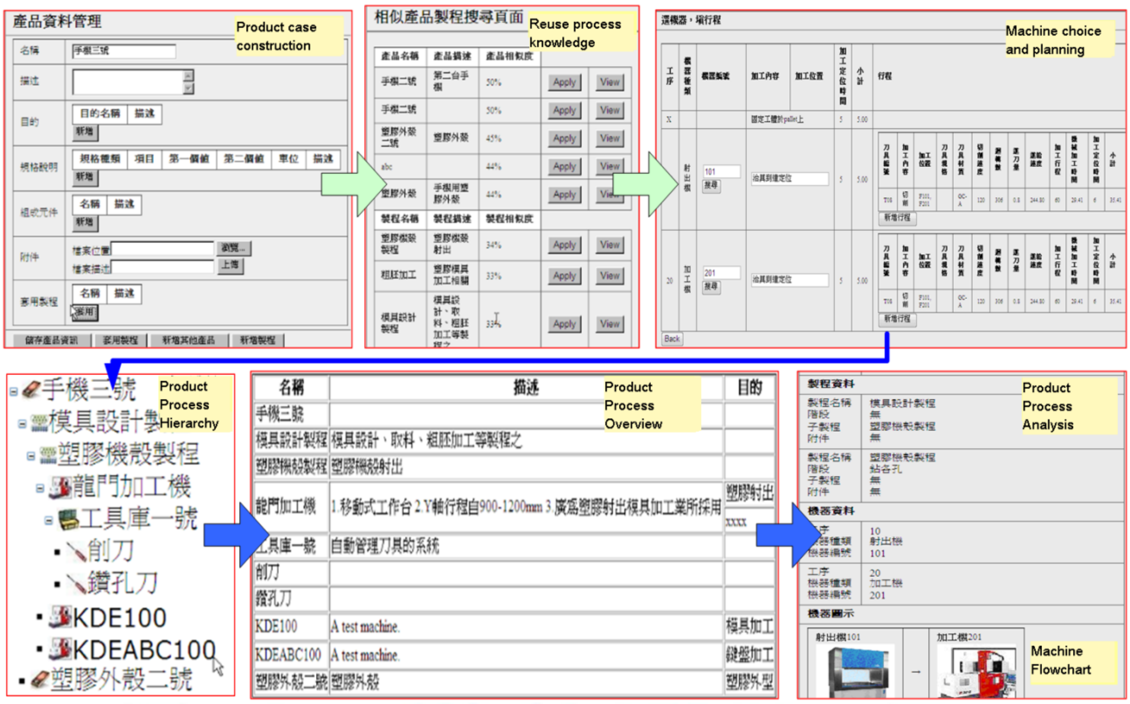


Figure 8: The process planning steps in the system in our experimental system

In Figure 8, we can see that there are six main steps to construct a new product and its process plan in our system, we introduce them below:

1. Product case construction

   In the beginning of new product construction, we have to fill out the requirements and specifications for this new product first.

2. Reuse process knowledge

   According to the requirements and specifications provided on previous stage, our system will execute some computation and then list all possible process case that might fulfill the requirements and specifications.

3. Machine choice and planning

   After appropriate process case is decided, product designer can start to choose applicable machines and adjust the sequences of the stages of actions that would be executed on these machines.

4. Representation of Product Process Hierarchy

   Then, our system will show the process planning generated by previous steps in a hierarchical representation. And this hierarchical representation can also be a hyper link to directly link to the processes. Product designers can quickly review and adjust any part of the process plan via this product process hierarchy.

5. Review of Product Process Overview

   This review shows a brief summarization of the process plan, including the chosen machines, and the sequences of actions executed on the chosen machines. Product can view and check if whole process plan is correct.

6. Outcome of Product Process Analysis and Flowchart

   After finishing above checking step, the system will provide a detail process plan analysis and also the machine flowchart. By this outcome, engineers can follow the flowchart and the analysis results to operate the machines and produce the new product.

Our system has been modified many times due to domain experts' opinions. And in the rest of this chapter, we discuss about some points for the HCBPP system provided by domain experts of mobile phone manufacturing.

6.2 Experiment Design

In this experiment, there are 13 mobile phone manufacturing domain experts who work for Industrial Technology Research Institute participated in the trials of our designed HCB-CAPP system for mobile phone manufacturing. There are many mistakes found in the exam paper.

Moreover, 11 computer science programmers who are familiar with the concept of case-based reasoning are also involved in this experiment to evaluate the effective and efficiency of our HCB-CAPP system for mobile phone manufacturing.

In the beginning, we show experts and programmers both the manual and the demo video. Then, we invite experts to use our system when they have to design a new product. On the other hand, we explain our system scheme and architecture and also present the effectiveness and advantages of our system to programmers. Finally, we design questionnaire based on some software evaluation metrics and request some mobile phone manufacturing domain experts and some programmers to evaluate our system performance.

6.3 Experiment Result and Discussion

Our questionnaire focuses on several software evaluation metrics, we discuss with experts and programmers about the content of our designed questionnaire to make it more adaptive to be used to evaluate our system.

In all of our questions in the questionnaire, we focus on two metrics most. One is the satisfaction degree of the users after using our HCB-CAPP system. And the other, the advancement degree of our HCB-CAPP system compared to other traditional case-based CAPP system. Here we apply typical five-level Likert scale in our questionnaire and the format of a five-level Likert item is listed in Table 2.

Table 2: The Format of a Five-Level Linkert Item

Degree	Meaning
1	Strongly disagree
2	Disagree
3	Neither agree nor disagree
4	Agree
5	Strongly agree

Finally, we choose 6 and 7 questions to estimate these two metrics respectively. We list the questions in Table 3 and Table 4.

Table 3: Question Items in Questionnaire for Satisfaction Degree

Question Items in Questionnaire for Satisfaction Degree	
Q1.	I feel satisfied with the way to operate this system.
Q2.	I feel satisfied with the representation method of the content.
Q3.	I feel satisfied with the Product Process Hierarchy provided in this system.
Q4.	I feel satisfied with the interactive process planning mechanism.
Q5.	I feel satisfied with the rule-assisted case retrieval mechanism.
Q6.	Totally, I feel satisfied with the assistance with the HCB-CAPP system.

Table 4: Question Items in Questionnaire for Advancement Degree

Question Items in Questionnaire for Advancement Degree	
Q1.	Hierarchy structure is easier to be understood than flat one.
Q2.	Interactive operation is more likely to fulfill product requirements than one shot approach.
Q3.	Retrieval with rules can be more precise than only mathematical formulas.
Q4.	Collaborative environment can save personnel cost.
Q5.	Collaborative environment can save time.
Q6.	Using the HCB-CAPP system can generate appropriate product process plan more easily than traditional case-based system.
Q7.	Totally, the HCB-CAPP system is better than traditional case-based system.

And the results of the questionnaire are listed below:

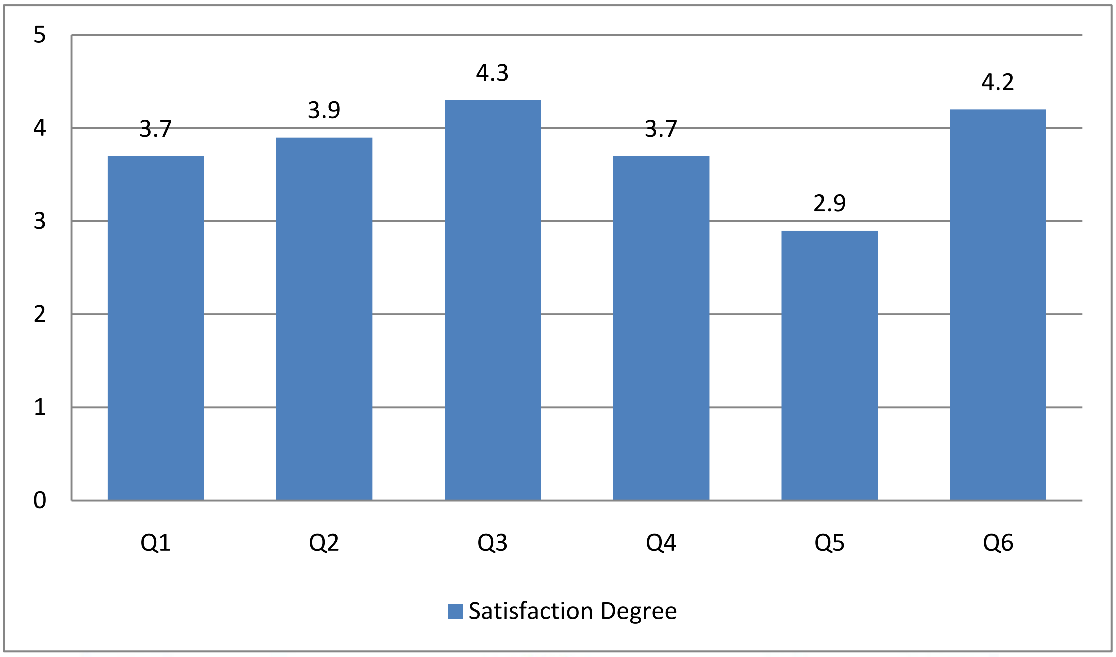


Figure 11: The Questionnaire Results of Satisfaction Degree

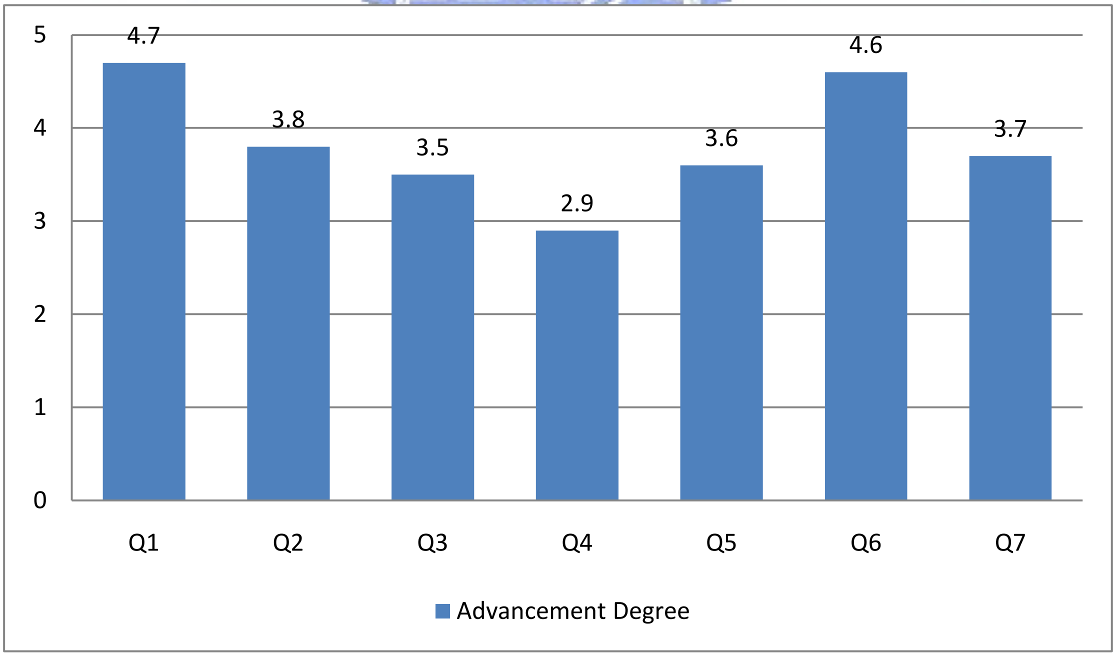


Figure 12: The Questionnaire Results of Satisfaction Degree

We discuss these results with experts and programmers and get some comments of them. Here we list these opinions below:

- **Advantage of HCBPP system over flat case structure systems**

A hierarchical case-based process planning system overcomes the disadvantages associated with previous systems which mostly use flat structure case-based approach. Some of advantages and distinguishing features of the proposed system are as follows:

1. Instead of generating solutions via single iteration, black box approach, hierarchical case-based system provide multi-level, multi-iteration, multiple revisions mechanism to generate solutions from coarse grain to fine grain. This leads to a solution that is closer to the new requirements and is more easier controlled by product designer.
2. For some domain knowledge with hierarchical naturally, hierarchical case-based are easy to maintain the domain knowledge. And the knowledge representation can be easy to be understood by domain experts.

- **An appropriate similarity function for specific data type**

The underlying intention in recognizing the most similar case is to minimize the adaptation of the process plan of the case to suit the part input by the user. The easier CBR systems used single equation for computing the similarity. In this experimental system, we can found it is very hard to find one similarity computing function to allow designers to have reasonable satisfaction due to low precision. So we choose multiple similarity computing functions to improve the precision of our retrieval result.

- **The importance of system interface**

Interface issues have always existed in computer applications, and especially in traditional case-based reasoning system due to the complex nature of the search task [19]. Our system applies an iterative input interface to facilitate the designers to input less information in each steps. But due to this mechanism, our system also provided more chances for designer to communicate information with system. So the interfaces of the system become a very important point that directly affects the satisfaction degree of designers.

In our interviews with the domain experts, they also remind us to consider more about the scenario of how designer interact with our HCBPP system and make the user interface as friendly as possible. If necessary, system engineer should try to get some feedbacks from product designers as a reference for system improvement.

- **Constrains of HCB-CAPP system**

The followings are the limitations associated with most of the CBR systems including the system we proposed in this research.

1. Most of the case-based systems are unable to maintain the required performance associated with similarity measurement, case retrieval and the case adaption, etc.
2. In general, for most of the CBR systems, robustness is directly proportional to the size of case libraries.
3. On the other hand, response time is also proportional to the size of case libraries.

Chapter 7. Conclusion

In the mobile phone manufacturing domain, apply CBR is suitable because this kind of system can fulfill the fast changing environment and mass production of short life cycle products. Many researchers have been proposed to manage and reuse previous process planning case. But traditional flat structure case-based approaches not only appear insufficient support in those domains with hierarchical characteristic but also often apply an “all requirements in, one solution out” black box approach. Using this kind of black box approach system, designer has to know all the details of the requirements, and might be difficult to modify the solution generated by the system due to lack of information about generating process. Our proposed hierarchical case-based process planning system can show the steps and allow designers to modify the inappropriate part or the requirements during generating phase.

According to the opinions and suggestions from several domain experts and product designer, the proposed system can assist the designer more than traditional flat structure case-based system. As a computer-assisted manufacturing process planning system, our system still involved too many manual operations, in the near future, the retrieval and adaptation mechanism will be extended to apply more automatic mechanism. Furthermore, more domain depend similarity measurement might generate solutions that match the thinking of expert s or designers more and therefore increase the satisfaction degree of system users.

References

[1] B.J. Davies, I.L. Darbyshire. The use of expert systems in process planning. Annals of the CIRP, Vol.33, n.1, p.303-306, 1984

[2] B. Smyth, M.T. Keane, P. Cunningham. Hierarchical Case-Based Reasoning Integrating Case-Based and Decompositional Problem-Solving Technique for Plant-Control Software Design, IEEE Transactions on Knowledge and Data Engineering, Vol.13, Issue.5, p.793-812, 2005

[3] C. Tsatsoulis, R.L. Kashyap. Case-based reasoning and learning in manufacturing with the TOLTEC planner, IEEE Transactions on Systems, Man and Cybernetics, Vol.23, Issue.4, p.1010-1023, July-August, 1993

[4] D. Hinkle, C. Toomey. Applying Case-Based Reasoning to Manufacturing, AI Magazine, Vol.16, n.1, p.65-73, 1995

[5] D.R. Liu, C.K. Ke, M.Y. Wu. Context-Based Knowledge Support for Problem-Solving by Rule-Inference and Case-Based Reasoning, 2008 International Conference on Machine Learning and Cybernetics, Vol.6, p.3205-3210, 2008

[6] D.S. Nau, M. Gray. SIPS: An application of hierarchical knowledge clustering to process planning. In Integrated and Intelligent Manufacturing, ed. C. R. Liu and T. C. Chang. American Society of Mechanical Engineers, p.219-225, 1986

[7] D.S. Nau, S.K. Gupta, W.C. Regli. AI planning verse manufacturing-operation planning: A case study. In Proceedings of the International Joint Conference on Artificial Intelligence. Montreal, Canada. Morgan Kaufmann Publishers, San Mateo, CA, p.1670-1676, 1995

[8] F.C. Yuan, C.C. Chiu. A hierarchical design of case-based reasoning in the balanced scorecard application, Expert Systems with Applications, Vol.36, Issue.1, p.333-342, January, 2009

[9] F.T.S. Chan. Application of a hybrid case-based reasoning approach in electroplating industry, Expert Systems with Applications, Vol.29, p.121–130, 2005

[10] G. Qiao, R. Lu, C. McLean. Flexible Manufacturing System for Mass Customization Manufacturing, International Journal of Mass Customisation, Vol.1, Issue.2/3, p.374 - 393, 2006

[11] H.T. Wang, and Y.T. Liu. Hierarchical Case-Based Decision Support System for Power System Restoration, Power Engineering Society General Meeting, Vol.1, p.1115–1119, 2004

[12] I. Watson, S. Perera. A hierarchical case representation using context guided retrieval, Knowledge-Based Systems, Vol.11, p.285–292, 1998

[13] J. Britanik, M. Marefat. Case-based manufacturing process planning with integrated support for knowledge sharing, Assembly and Task Planning, p.107–112, 1995

[14] J. Dong, H.R. Parsaei, A. Kumar. Intelligent feature extraction for concurrent design and manufacturing. In Design and Implementation of Intelligent Manufacturing Systems, Chap. 13, ed. H. Parsaei and M. Jamshidi. Prentice-Hall, Upper Saddle River, NJ, p.301-323, 1995

[15] J.M. Usher. Object oriented approach to feature-based process planning. In Design and Implementation of Intelligent Manufacturing Systems, Chap. 12, ed. H. Parsaei and M. Jamshidi. Prentice-Hall, Upper Saddle River, NJ, p. 275-300, 1995

[16] K. E. Hummel. Coupling rule-based and object-oriented programming for the classification of machined features. In Proceedings of the ASME Computers in Engineering Conference, San Francisco, CA, p.409-418, 1989

[17] M.K. Tiwari, K.R. Kotaiah, S. Bhatnagar. A Case-Based Computer-Aided Process-Planning System for Machining Prismatic Components, The International Journal of Advanced Manufacturing Technology (2001), Chap.17, p.400–411, 2001

[18] P. Veerakamolmal, S.M. Gupta. Automating Multiple Products Disassembly Process Planning with Case-Based Reasoning, The Second International Conference on Operations and Quantitative Management (ICOQM-II) January 3-6, Ahmedabad, India, 1999

[19] W. He, F.K. Wang, T. Means, L.D. Xu. Insight into interface design of web-based case-based reasoning retrieval systems, Expert Systems with Applications, Vol.36, n.3, p.7280-7287, April, 2009

[20] Y. Descotte, J.C. Latombe. “GARI: a problem solver that plans to machine mechanical parts”, Proceedings of 7th International Joint Conference on Artificial Intelligence, p.766-772, 1981

[21] Y.F. Li, M. Xie, T.N. Goh. A study of mutual information based feature selection for case based reasoning in software cost estimation, Expert Systems with Applications, Vol.36, Issue.3, p.7280-7287, April, 2009