

私立東海大學  
資訊工程與科學所

碩士論文

指導教授：朱正忠 博士

以模糊的目標-問題-度量指標方法論為基礎之軟

體度量規劃系統

A Fuzzy GQM based Software Measurement  
System

研究生：林致偉

中華民國九十三年六月

# 摘要

在軟體需求與日俱增的情形下，軟體的規模由小變大，系統複雜度由簡單變複雜，面對這種需求的發展，軟體設計構思不再單純只是其開發程度的完全與否，品質的議題已開始受到重視。在軟體品質的控管上，首先有軟體度量(Measurement)的思維出現，透過度量的方式，取得的資料間接反映出專案在開發中或者是維護中相關的重要訊息，可想而知，以管理的層面而言，可根據客觀的度量結果而做出合適的決策。

度量規劃運用目標-問題-度量指標方法論(Goal/Question/Metric, GQM)此方法來輔助進行有一定的幫助，因為度量指標相當繁多，要「指出專案所欲獲得的資訊」並不容易，接下來的推論度量指標自然也就更為困難，所以，如果將 GQM 的方法運用到度量流程的規劃階段，便能有方向性的去選擇需要的度量指標。

在本論文中，我們將提出一個以 Fuzzy GQM 為基礎之軟體度量規劃系統(Fuzzy GQM Based Software Measurement System)。雖然 GQM 在理論上能有效的輔助度量規劃人員進行規劃，但其實仍有其應用上的問題。倘若度量規劃人員的度量經驗不足夠，且對軟體開發的程序並不十分了解，即使 GQM 提供了一種選擇度量指標的方法，仍然無法解決經驗不足的問題。本系統便是一套將 GQM 理論在應用上的問題所改良之度量規劃輔助系統，利用知識庫紀錄過往度量規劃的專案資料，運用模糊理論的概念定義目標、問題以及度量指標之間的關係，並利用 Fuzzy Petri Net 來儲存以及表現 GQM 關係，將目標、問題以及度量指標以動態的方式供度量規劃人員選擇，以此協助度量規劃經驗不足的規劃人員進行度量規劃。Fuzzy Petri Net 已被廣泛的運用在許多的知識庫系統，他能夠以結構化的方式表達知識以及 Fuzzy 推論規則。

關鍵字：軟體度量、度量指標、目標-問題-度量指標方法論、Fuzzy Petri Net

## Abstract

The continuing improvement of software quality is extremely important for software developments. Measures are the fault detectors that let us know when our projects and processes encounter problems. Further, measures can give us an objective indication whether an improvement actions working or not. To accomplish quality in software processes and products, it is required to develop systematic measurement programs, congenial with the organizational purpose and property and tailored to the quality aspects that are being advised.

The Goal-Question-Metric (GQM) approach was proposed by Victor Basili et. al.. The main purpose of the GQM is to guide the definition and development of a goal-driven measurement program.

Although the GQM seems to be simple to apply, it involves a sequence of well defined, relative stages, phases and activities that are not easy to understand and apply for non-expert. In this paper, we proposed a Fuzzy GQM Based Software Measurement System. This system applies fuzzy Petri net to represent the fuzzy GQM relation. Exploiting such mechanism, our system can dynamically provide goal list, question list and metric list for users depend on specific project features. Users only need to choose from options of goals and questions, a list of metrics will be recommended by the decision support system. Moreover, this system enables to capture experiences in the knowledge base so that such experiences could be reused in other projects. The Fuzzy Petri-Net is adopted for the analysis the relationship between goal, question, and metric in this system.

**Keywords:** software measurement, Metrics, GQM, Fuzzy Petri net

# 目錄

摘要.....	I
ABSTRACT .....	II
目錄.....	III
圖目錄.....	V
表目錄.....	VII
<b>第一章 前言 .....</b>	<b>1</b>
1.1 研究動機與目的 .....	1
1.2 研究流程與架構 .....	3
1.3 章節安排 .....	4
<b>第二章 背景知識及相關研究 .....</b>	<b>5</b>
2.1 ISO/IEC STD 15939 .....	5
2.2 目標-問題-度量指標方法論(GOAL/QUESTION/METRIC, GQM) .....	7
2.3 模糊理論 .....	9
2.3.1 模糊理論簡介.....	9
2.3.2 模糊推論決策行為理論基礎.....	10
2.4 FUZZY PETRI NET .....	11
<b>第三章 模糊的目標-問題-度量指標方法論.....</b>	<b>13</b>
3.1 模糊 GQM 關係 .....	13
3.1.1 GQM 理論 .....	14
3.1.2 Fuzzy Petri Net 與模糊 GQM 關係的表現與分析.....	15
3.2 模糊專案分類 .....	18

<b>第四章 系統架構</b> .....	<b>22</b>
4.1 知識庫與 GQM 處理器 .....	24
4.1.1 度量目標、度量問題與度量指標的關係.....	25
4.1.2 新增度量目標、度量問題、度量指標的關係.....	27
4.2 度量指標資料庫與 GQM 處理器 .....	28
4.2.1 傳送度量指標資訊.....	28
4.2.2 新增度量指標.....	28
<b>第五章 系統執行流程以及案例探討</b> .....	<b>30</b>
5.1 系統執行流程 .....	30
5.1.1 度量規劃流程.....	30
5.1.2 更新度量知識庫流程.....	31
5.2 案例探討 .....	33
<b>第六章 系統設計與系統畫面</b> .....	<b>39</b>
6.1 USE CASE: SELECT SYSTEM .....	41
6.2 USE CASE: INPUT PROJECT FEATURE.....	43
6.3 USE CASE: OUTPUT GQM OPTION LIST AND METRIC DETAIL INFORMATION .....	45
6.4 USE CASE: SELECT GQM OPTION.....	47
6.5 USE CASE: SELECT UPDATE TARGET .....	48
6.6 USE CASE: RECEIVE NEW DATA.....	49
6.7 USE CASE: UPDATE KNOWLEDGE-BASE.....	51
<b>第七章 結論與未來方向</b> .....	<b>53</b>
<b>第八章 參考文獻</b> .....	<b>55</b>

# 圖目錄

圖 1、軟體度量流程[8].....	6
圖 2 模糊決策架構圖[23].....	10
圖 3、驅動前的 Fuzzy Petri Net .....	17
圖 4、驅動後的 Fuzzy Petri Net .....	17
圖 5、Example of Fuzzy Petri Net.....	18
圖 6、Membership Function of N, S, and T .....	20
圖 7、專案特徵值分類範例.....	20
圖 8、專案規模分類模糊規則表.....	21
圖 9、度量工具與軟體開發流程的關係示意圖.....	22
圖 10、系統架構圖.....	23
圖 11、知識庫與 GQM 處理器之內部架構與關係示意圖 .....	25
圖 12、GQM 關係資料結構示意圖 .....	26
圖 13、GQM 關係分析器處理示意圖 .....	27
圖 14、度量指標資料庫與 GQM 處理器的關係與內部架構示意圖 .....	28
圖 15、FGQMSM 度量規劃流程圖 .....	31
圖 16、知識庫更新流程圖.....	32
圖 17、Example of Fuzzification.....	33
圖 18、Example of GQM Fuzzy Petri net .....	35
圖 19、Example of relations between Questions and Metrics.....	36
圖 20、系統 Use Case Diagram.....	39
圖 21、選擇系統畫面.....	41
圖 22、度量規劃系統之初始畫面.....	42

圖 23、更新系統知識庫之初始畫面.....	42
圖 24、專案特徵值輸入位置.....	43
圖 25、專案分類後之結果顯示.....	44
圖 26、Input Project Feature 之 Activity Diagram.....	44
圖 27、度量目標選單.....	45
圖 28、度量問題選單.....	46
圖 29、度量指標選單.....	46
圖 30、度量指標詳細資訊列表.....	47
圖 31、Output GQM Option List and Metric Detail Information 以及 Select GQM Option 之 Activity Diagram .....	48
圖 32、可供選擇的知識庫更新項目.....	49
圖 33、模糊 GQM 關係的更新輸入畫面 .....	50
圖 34、歸屬函數的更新輸入畫面.....	50
圖 35、專案分類規則的更新輸入畫面.....	51
圖 36、Receive New Data 和 Update Knowledge-Base 的 Activity Diagram .....	52

# 表目錄

表格 1、一個以 GQM 作度量規劃之例子 .....	14
表格 2、專案特徵值[38].....	19
表格 3、度量指標相關資訊[39].....	29
表格 4、Example of the Project Features .....	34
表格 5、Example of the project classified output.....	34
表格 6、Example of Goal list .....	36
表格 7、Example of Question list.....	36
表格 8、Example of Metric list .....	37
表格 9、Example of Metric Information .....	37
表格 10、Compare fuzzily GQM relation with crisp GQM relation.....	37
表格 11、Use Case 定義.....	40

# 第一章 前言

## 1.1 研究動機與目的

在軟體需求與日俱增的情形下，軟體的規模由小變大，系統複雜度由簡單變複雜，面對這種需求的發展，軟體設計構思不再單純只是其開發程度的完全與否，相反的，品質的議題開始受到重視，1997年由美國國防部委託 Carnegie Mellon University 軟體工程學院(Software Engineering Institute, SEI)提出整合能力成熟度模式(Capability Maturity Model-Integrated, CMMI ) [1]，成為國際間認可的軟體生產流程標準，說明了軟體品質的重要性已受到許多研究學者的重視，軟體品質研究的議題成為二十世紀末的一個發展主流。

ISO 9001 [2]是工業上用來控管品質的標準，相對的，在軟體產業尚未有一明確的標準出現，也因為如此，讓許多人想要維護軟體品質有心有餘而力不足的情況發生，因此，在軟體品質的控管上，首先有軟體度量(Measurement)的思維出現，透過度量的方式，取得的資料間接反映出專案在開發中或者是維護中相關的重要訊息，可想而知，以管理的層面而言，根據客觀的度量結果而做出合適的決策，有其度量的意義所在，執行度量的優點包含下列幾點：

### 1. 有效的溝通：

透過度量的方式，讓參與專案的人員能夠對專案更了解，減少人員彼此之間的誤解。

### 2. 追蹤專案狀況：

根據度量結果來追蹤專案進行的狀況，舉凡重要性的問題，如“專案是否如排程進行？”、“專案是否能如期結案移交給客戶？”。

### 3. 早期發現與修正問題：

經由度量結果的發現，讓專案管理者能夠早期發現不正常的問題，進而早期修正，避免在專案後期才發現而須付出更多成本的風險。

### 4. 利益取捨的決策：

一般專案的開發，常常會遇到成本、時程等壓力，因此在決定決策時，度量結果可以作為一個參考依據。

### 5. 決策的驗證：

專案管理者對於專案的決策，可以透過度量的結果來驗證專案管理者的決策是否合適，經由雙重的驗證以確保決策的適宜性。

一個軟體度量程序中，度量規劃最為重要也最為困難，包括以下三項工作：

1. 指出專案所欲獲得的資訊、2. 選擇可獲得專案所需要的資訊之度量指標、3. 整合度量程序與軟體開發程序，使得軟體開發程序能與度量程序配合，以便收集度量所需之資料並進行分析。其中，「指出專案所欲獲得的資訊」以及「選擇可獲得專案所需要的資訊之度量指標」兩項工作，運用目標-問題-度量指標方法論 (Goal/Question/Metric, GQM)[3][4][5]此方法來輔助進行是有一定幫助的，因為度量指標相當繁多，要「指出專案所欲獲得的資訊」並不容易，接下來的推論度量指標自然也就更為困難，所以，如果將 GQM 的方法運用到度量流程的規劃階段，便能更有方向性的去選擇需要的度量指標。

在本論文中，我們將提出一個以 GQM 為基礎的度量決策支援系統(Fuzzy GQM Based Software Measurement System)。雖然 GQM 在理論上能有效的輔助度量規劃人員進行規劃，但其實仍有其應用上的問題。倘若度量規劃人員的度量經驗並不足夠，且對軟體開發的程序並不十分了解，即使 GQM 提供了一種選擇度量指標的方法，仍然無法解決經驗不足的問題。本系統便是一套將 GQM 理論在應用上的問題所改良之度量規劃輔助系統，利用知識庫紀錄過往度量規劃的專案

資料，定義目標、問題以及度量指標之間的關係，並將目標、問題以及度量指標以選單的方式供度量規劃人員選擇，以此協助度量規劃經驗不足的規劃人員進行度量規劃。

在定義目標、問題以及度量指標之間的關係時，本論文運用了模糊理論的概念定義目標、問題以及度量指標之間的關係，並利用 Fuzzy Petri Net 來儲存以及表現 GQM 關係，藉此提升 GQM 關係的彈性以及實用性。Fuzzy Petri Net 已被廣泛的運用在許多的知識庫系統，能夠有效的利用結構化方式表達知識以及 Fuzzy 推論規則。[6][7]

## 1.2 研究流程與架構

研究流程係根據研究目的及方法架構其流程，並分為下列幾項說明：

### 1. 界定研究問題、目的與範圍

針對目前已存在之軟體度量流程進行了解，並分析各軟體度量流程中度量規劃步驟之特性以及相關重點。

### 2. 目標-問題-度量指標方法論與模糊理論之理論分析

在本研究中，目標-問題-度量指標方法論以及模糊理論是兩大研究重點，因此，將對此兩項理論進行更深入的分析與探討。

### 3. 模糊理論與目標-問題-度量指標方法論之整合

將模糊理論與目標-問題-度量指標方法論進行整合，並以此規劃模糊目標-問題-度量指標推論系統之系統架構。

### 4. 模糊目標-問題-度量指標決策支援系統之構建

根據本研究的系統規劃，進行系統實做，以此驗證本研究所提出之理論的可

行性。

### 1.3 章節安排

本論文章節架構如下：

第一章 前言：介紹本文研究動機及研究目的。

第二章 背景知識及相關研究：介紹與本文有關之相關理論與技術。

第三章 系統架構：針對本系統的核心架構以及處理流程做詳細介紹。

第四章 度量指標知識庫：獨立介紹本系統中，最重要的度量指標知識庫

第五章 系統執行流程以及案例探討：利用一個案例介紹本系統之執行流程。

第六章 系統設計與系統畫面：系統最後的實作畫面。

第七章 結論與未來方向：將研究成果歸納出具體結論，並提出未來後續研究之建議

## 第二章 背景知識及相關研究

### 2.1 ISO/IEC std 15939

軟體度量流程的主要目的在於透過軟體度量指標對軟體開發專案進行度量，獲得專案的相關資訊，以此來控管軟體開發專案的品質與效率。目前軟體度量流程的相關研究與標準相當的多，其中尤以 ISO/IEC std 15939[8][9]，軟體工程-軟體度量流程( Software Engineering - Software Measurement Process)此國際標準的定義最廣被認同，許多國際學者的研究主題也是以此標準作為基礎[10][11]。主要的研究方向以定義與改善軟體度量流程為主，如下圖 1 所示，首先在建立並維護度量計畫(Establish Commit)階段，接受開發專案的度量需求與度量所需資源，在規劃階段(Plan)與執行階段(Perform)則對度量需求進行分析、選擇適用的度量指標並進行度量，最後再將度量的結果傳遞給專案管理人員與評估階段(Evaluate)進行軟體度量流程的改良。

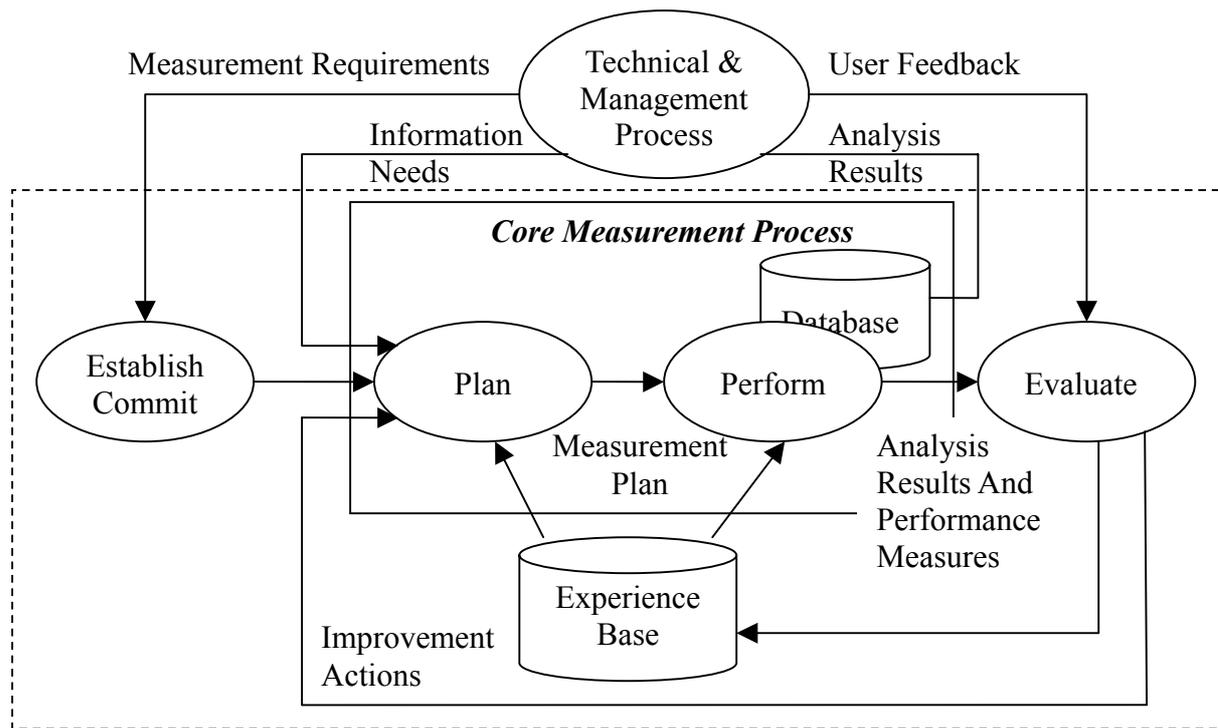


圖 1、軟體度量流程[8]

其中度量流程中的四個主要部分之工作敘述如下：

(1) 建立並維護度量計畫(Establish Commit)

- a. 接受並分析度量需求
- b. 指定以及分派資源

(2) 規劃度量計畫(Plan)

- a. 對組織單元進行分析與描述
- b. 指出資訊需求
- c. 選擇度量指標
- d. 定義資料收集、分析與報表的程序
- e. 定義評估資訊產品與度量程序的標準
- f. 分配支援至各度量工作
- g. 決定並分派支援度量的技術

### (3) 執行度量工作(Perform)

- a. 整合需執行的度量工作
- b. 收集需度量的資料
- c. 分析資料並產出報表
- d. 傳遞結果至相關單位

### (4) 評估度量成效(Evaluate)

- a. 評估度量後所獲得的資料以及度量的程序
- b. 指出可以改進的項目

## 2.2 目標-問題-度量指標方法論(Goal/Question/Metric, GQM)

目標-問題-度量指標方法論(Goal/Question/Metrics, GQM)是由 Basili 與 Rombach 兩位學者於 1988 年所發展，現在已成為軟體度量的重要方法論之一。由於此概念並不只限於專案規劃，因而被應用到許多不同的領域當中，包括軟體度量、軟體開發流程改善、工業上的品質管理等等[12][13][14]。

GQM 將專案欲完成的工作分成三個步驟來考慮，首先是 Goals，目標，可以解釋為專案成員所欲達到或完成的目標，第二個是 Questions，問題，也就是為了達到 Goal 所會產生的問題，最後是 Metrics，度量，選擇正確的度量指標進行度量，並利用度量得到的結果來回答 Questions，當所有的 Questions 都被解決後，Goal 也就達成了。GQM 的主要步驟如下：

#### (1) 設定品質目標

目標可針對涵蓋整個過程發展、產品輸出、以及支援分配之專案任務的任一特性而設定，而且目標要簡明扼要，以便對於目前的發展狀況能更深入理解，並明確指出要評估或改善哪些品質重點。

#### (2) 設計問題

針對每一個目標，推導提出一些相關問題，以顯示上數之品質目標是否已經達成。

### (3) 規劃度量項目

將前述之問題，轉換為易於回答的量度項目(metric)，並開始規劃資料的收集和分析。

雖然目標-問題-度量指標方法論在軟體專案的度量與管理上，佔有相當重要的一席，不過，GQM 在應用上仍存在著許多問題，由於 GQM 只是一種輔助規劃的工具，如果規劃人員的經驗不足，對該領域的知識背景不夠，即使運用 GQM 來協助，仍無法達到很好的規劃成效。因此，為了克服 GQM 在應用上的問題，本論文提出了一套改良 GQM 的度量規劃系統，此系統利用知識庫儲存過往的度量規劃經驗，並配合 GQM 的理論基礎，可有效的改善 GQM 的問題並保留 GQM 的優點。

近年來，亦有許多學者開始對 GQM 方法論進行改良。如 MacDonell[15]學者的研究中，他們將 GQM 方法論做進一步的延伸，在其中再引入了子目標(Sub-goal)、領域(Domain)、子領域(Sub-domain)、子問題(Sub-question)與特徵度量(Characteristic Measure)等概念，藉由對不同領域的分類，讓問題更貼切於相對的領域，進而使問題的解決更加明確。此外，依據領域的差異，對問題也可以做不同的詮釋，使該領域的人員可以更容易了解與執行。

而 Gray[16]等學者的研究則是將 MacDonell 學者的研究再加入三個元件於度量指標內，使整個度量活動的實施更加明確，且可以更容易地發現問題點的存在。而三個元件分別敘述如下：

- (1) 資料來源(Data Source)：定義該度量指標的資料蒐集方式、資料格式與資料來源等。
- (2) 技術(Technique)：定義模式建構的技術。
- (3) 執行(Implementation)：定義相關模式分析的相關執行方式。

最後，Olsson[17]等學者的研究則是將驗證的機制引入目標－問題－度量指

標 GQM 方法論，以增加模式的精確性與可靠性。

在 Niessink and van Vliet[18]的研究中定義了一個將商業以及組織目標納入的 GQM 規劃流程。在 Birk et. Al.[19]的研究中，提出了一個延伸管理度量目標的方法，並使用了數個例子作驗證。另外，在 Offen 以及 Jeffery[20]的研究中，將 GQM 作了延伸，他們多增加了幾個步驟在標準的 GQM 方法中。

## 2.3 模糊理論

模糊理論是將語言概念的「模糊性」、評價判斷的「模糊性」，以模糊(Fuzzy)集合做數學處理的理論。在許多方面，模糊數學已成功地幫助人們處理具不明確訊息的知識領域，諸如醫學、社會經濟、化學製程、與環境工程、自動控制等，並改進了專家在經驗判定的評價優劣。[21][22]

### 2.3.1 模糊理論簡介

#### (1) 模糊集合與一般集合之比較

模糊集合較於一般集合論，其最大之差異在於描述一事件之歸屬，或是非之方式。一般集合論特色之一即是排中律，或稱為二分法，亦即一事件屬於某一集合或不屬於某一集合，二者必只有一成立。其隸屬度不是 1 就是 0，因此其特性劃分十分明確。然而人類在語言上的表達，以及思想經驗或概念上的表達都是相當模糊的，因此 Zadeh[21]教授於 1965 年所提出的模糊理論，即是建立一套語言分析的數學模式，將那些模糊不清的事物予以明確化、數據化，進而轉成機器所能接受的運算語言。亦即模糊邏輯是將傳統之邏輯值 0 與 1 間建立一緩衝地帶，以較具彈性的空間，來表達一些概念的理念與經驗。

#### (2) 明確集合、模糊集合與隸屬函數

- a. 明確集合(Crisp Set)：定義在論域(Universe of Discourse)  $Z=\{X\}$  上的一個集合A，若

$$u_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

其中  $u_A(x)$ ：特徵函數(Characteristic Function)

則  $A$  為一明確集合。

b. 隸屬函數(Membership Function)：為明確集合中特徵函數一般化形式，它把全集  $U$  映射到閉區間  $[0,1]$  上去，賦予每一個元素一個介於 0 到 1 之間的數值（包括 0 與 1 在內）。 $\mu$  值愈接近 1，則  $\mu$  屬於 Fuzzy 集合  $A$  的程度越高，越接近 0，則表示越低。

c. 模糊集合(Fuzzy Set)：是以界線不清楚事物的集合體為對象。模糊集合如以數學來定義，則變成全集  $U$  所在的模糊集合(Fuzzy set)  $A$  為：

$$\mu_A : U \rightarrow [0,1]$$

以隸屬函數(membership function)  $\mu_A$  表示。

### 2.3.2 模糊推論決策行為理論基礎

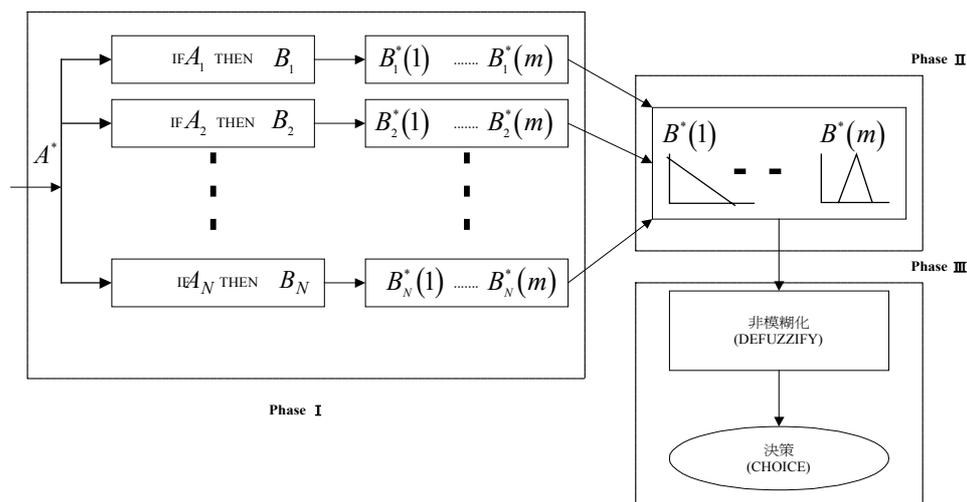


圖 2 模糊決策架構圖[23]

根據模糊決策架構，可以其過程分為三個處理階段[23]如圖 2，分述如下：

第一階段：近似推論階段(The Approximate Reasoning Phase)：輸入屬性變數，透過 IF..then 的決策規則之推演及近似推論方式，產生每一規則下各方案的初步推論結果。

第二階段—內部處理表示階段：在模糊決策中之規則具有彈性，可與實際環境加以結合，且可考慮多規則之組合，因此必須透過加總機制(aggregation scheme)來反應最後之決策反應。針對各決策規則的推論結果加以結合，產生各方案之模糊評價，用來作為選擇集合中各方案的偏好程度。

第三階段—非模糊化階段：由於方案的決策為一離散選擇，因此將選擇集合中各方案的模糊偏好做一比較，偏好最高的方案將會被決策者所選擇。

## 2.4 Fuzzy Petri Net

為了恰當的去表達真實世界的知識，模糊推論規則被應用於陳述這些知識[24][25][26]，模糊推論規則陳述兩個物件或理念之間的模糊關係。像這樣的模糊推論規則，可以使用 Fuzzy Petri Net[27]去表達的更為清楚。Fuzzy Petri Net 為一種雙向圖(bipartite directed graph)，由位置(place)和轉置(transition)兩種資料型態組成，圓形表示位置，而條形代表轉置。每個位置可能包含了一個零到壹之間的信賴值，而轉置也有一個零到壹之間的可信賴程度，兩種節點之間的關係以弧線(directed arc)來表示。

在本論文中，利用 Fuzzy Petri Net 來分析及表現模糊 GQM 關係。GQM 的核心理論即是以度量目標、度量問題以及度量指標三者的推論關係來幫助規劃人員選擇度量指標。但是同樣的度量目標所對應的度量問題並非每次都是相同的，會因為不同的專案類型以及不同的專案需求而有所不同，因此，在本論文中所提出的度量規劃輔助系統中，知識庫所儲存的度量規劃經驗利用了模糊理論之觀念

，將模糊的元素加入了 GQM 關係中，藉此提升本系統的適用性及正確性。

許多的學者亦將研究重點放在 Fuzzy Petri Net。在[27]中，Chen 提出了一個 Fuzzy Petri Net 模式來表達知識以及模糊關係。在[28]中，Peters 將 Fuzzy Petri Net 加上了顏色，並應用於表現平行程式，藉此模擬專家系統的計算。在[29]中，Zhou 以及 Murata 提出了一個加上時間以及模糊即時邏輯的 Petri net 模式。在[30]中，Chen 擴充了[27]中的研究，提出了一個加上權重值的 Fuzzy Petri Net 模式，並提出了加上權重值的模糊推論演算法。在[31]中，Knoar 以及 Mandal 提出一種方法解決在專家系統使用 Fuzzy Petri Nets 過程中的不確定管理。

在[32]中，Looney 提出了一個應用於以 Fuzzy Petri Nets 的決策支援系統之演算法。在[33]中，Scarpelli 提出了一個使用 High-Level Fuzzy Petri Nets 的模糊推理演算法。在[34]中，Yeung 以及 Tsang 提出了一個多階層的權重模糊推論演算法，應用於以 Fuzzy Petri Nets 為核心的專家系統。

在[35]中，Carinena 介紹了一個增強模糊暫時知識庫形式的模式，此模式將模糊的暫時規則利用 Fuzzy Petri Nets 的規範來呈現。在[36]中，Wu 提出了一個 Fuzzy Timing Colored Petri Net 模式，並且討論在某些特定狀況下，此模式可能發生的事件。

# 第三章 模糊的目標-問題-度量指標

## 方法論

本論文採用目標-問題-度量指標方法論(Goal/Question/Metrics, GQM)作為核心的度量規劃理論，因此，在度量知識庫中所儲存的度量經驗即為 GQM 關係，係指度量目標、度量問題與度量指標之間的對應關係，但是每個 GQM 對應關係中度量目標、度量問題與度量指標之間的對應情形並非是絕對的，也就是說，不同專案規劃人員對於同一個度量目標所對應的度量問題會有不同看法。基於此點，本論文在 GQM 關係中，加入了模糊理論的元素，將原本固定的 GQM 關係轉換為模糊的動態 GQM 關係，以適應不同的專案類型並調整度量經驗的獲得，而這樣的模糊關係相當適合以 Fuzzy Petri Net 來呈現與分析，因此，本論文中便以 Fuzzy Petri Net 來表現模糊 GQM 關係。

另外，如能在對知識庫查詢度量經驗之前，預先對專案進行分類處理，可以有效地提高從知識庫中獲得度量經驗之準確性，因此，在本論文中，亦對專案分類進行研究，透過使用者所輸入的專案特徵值進行專案分類，並加入了模糊理論的元素，降低度量規劃人員在輸入專案特徵值時的困難。

在本章中，將針對上述兩個理論進行詳細的敘述。

### 3.1 模糊 GQM 關係

在本章節中，將對 GQM 此理論進行介紹，並以一個例子表現 GQM 應用於度量規劃時的情形。另外，亦將介紹如何將模糊理論與 GQM 理論做結合，形成本論文所提出的模糊 GQM 關係。

### 3.1.1 GQM理論

由於在本論文的 2.2 章節中，已對 GQM 理論有所介紹，因此，在本章節，將不再贅述，只針對 GQM 在軟體度量規劃的應用上做加強敘述，並以一個簡單的例子做介紹。

GQM 將專案欲完成的工作分成三個步驟來考慮，分別是 Goals，目標，可以解釋為專案成員所欲達到或完成的，第二個是 Questions，問題，也就是為了達到 Goal 所會產生的問題，最後是 Metrics，度量指標，選擇正確的度量指標進行度量，並利用度量得到的結果來回答 Questions，當所有的 Questions 都被解決後，Goal 也就達成了。

表格 1 為一個 GQM 在軟體度量規劃的例子，其中，目標為「改善處理程序改變時所造成的閒置時間，從管理者的角度」，分別衍生出了兩個問題，分別是「目前改變處理程序時所造成的閒置時間為多少？」以及「改進後的效率如何？」，並且各自對應了一個度量項目。其中，較為重要的為目標這個部分，從表格中可以看出目標的敘述共分成了四個部分，目的(purpose)、議題(Issue)、對象(Object)、觀點(Viewpoint)。將目標以這樣的格式去敘述是為了要幫助解析使用者的語意，以便進行接下來的設計問題以及規劃度量項目。

表格 1、一個以 GQM 作度量規劃之例子

Goal Purpose	Characterization
Issue	the product quality
Object (process)	of specific method
Viewpoint	from the project manager's viewpoint
Question	Were the readability properties of the product any different?
Metric	Amount of comments
Question	Was the control complexity any different?
Metric	Measure the complexity

GQM 在理論上確實能幫助度量規劃人員一步一步的完成度量規劃工作，不過，GQM 在應用上仍存在著許多問題，由於 GQM 只是一種輔助規劃的工具，如果規劃人員的經驗不足，或許在 GQM 中的第一步驟訂定度量目標，便無法順利完成，更別提到接下來的推論度量問題以及度量指標了。因此，為了克服 GQM 在應用上的問題，本論文提出了一個改良 GQM 理論的度量規劃系統，此系統利用知識庫儲存過往的度量規劃經驗，並配合 GQM 的理論基礎，將度量規劃經驗轉換成為度量目標、度量問題以及度量指標選單，度量規劃人員便只需要從系統所提供的選單中選擇符合需求的選項，即可完成度量指標的選擇，如此一來，便可有效的改善 GQM 的問題並保留 GQM 的優點。

### 3.1.2 Fuzzy Petri Net與模糊GQM關係的表現與分析

本論文採用 GQM 作為軟體度量規劃的核心理論，因此，知識庫中所儲存的度量經驗即是過往以 GQM 作為度量規劃的經驗，也就是 GQM 關係。不過，每位度量規劃人員以 GQM 進行度量規劃時，對於度量目標、度量問題以及度量指標之間的對應關係看法並不一定一致，也就是說，不同專案規劃人員對於同一個度量目標所對應的度量問題會有不同看法。因此，本論文在 GQM 關係中，加入了模糊理論的元素，將原本固定的 GQM 關係轉換為模糊的動態 GQM 關係，以適應不同的專案類型與度量規劃人員需求，並調整度量經驗的獲得。

基於上述的原因，本系統採用 Fuzzy Petri net 來表現模糊 GQM 關係的模糊規則。Fuzzy Petri Net 不但可以有效的表達模糊的概念，更能清楚表現兩個位置(place)之間的關係，與 GQM 關係中的度量目標與度量問題之間的關係，以及度量問題與度量指標之間的關係相當符合，因此非常適合與 GQM 關係結合，故本系統將每一個 GQM 關係轉換為模糊的 GQM 關係，並利用 Fuzzy Petri Net 進行儲存與分析。

Fuzzy Petri Net 為一種雙向圖(bipartite directed graph)，由位置(place)和轉置

(transition)兩種資料型態組成，圓形表示位置，而條形代表轉置。每個位置可能包含了一個零到壹之間的信賴值，而轉置也有一個零到壹之間的可信賴程度，兩種節點之間的關係以弧線(directed arc)來表示。一個典型的 Fuzzy Petri Net 是由八個元素(8-tuple)所構成的集合：

$$FPN = \{P, T, D, I, O, f, \alpha, \beta\}$$

其個別定義如下：

$P = \{P_1, P_2, P_3, \dots, P_m\}$  為有限位置節點所構成的集合， $m$  表示在 Fuzzy Petri Net 中位置節點的個數， $m \geq 1$ 。

$T = \{T_1, T_2, T_3, \dots, T_n\}$  為有限轉置節點所構成的集合， $n$  表示在 Fuzzy Petri Net 中轉移節點的個數， $n \geq 1$ 。

$D = \{D_1, D_2, D_3, \dots, D_m\}$  為有限命題(domain)所構成的集合。

$$P \cap T \cap D = \phi \quad \text{且} \quad |P| = |D|$$

$I =$  為輸入位置節點(input place)所形成的集合。

$O =$  為輸出位置節點(output place)所形成的集合。

$f =$  為轉置節點的一個零到壹之間之可信賴程度。

$\alpha =$  為位置節點的一個零到壹之間之信賴度。

$\beta =$  為位置節點與命題的一對一對應函數。

假設  $A$  唯一有向弧線的集合。如果  $P_j \in I(T_i)$ ，則存在一個從位置  $P_j$  到轉置  $T_i$  的弧線  $a_{ji}$ ， $a_{ji} \in A$ 。如果  $P_k \in O(T_i)$ ，則存在一個從位置  $T_i$  到位置  $P_k$  的弧線  $a_{ik}$ ， $a_{ik} \in A$ 。若  $f(T_i) = \mu_i$ ，則  $T_i$  的可信賴程度等於  $\mu_i$ ；若  $\beta(P_i) = D_i$ ， $D_i \in D$ ，則  $P_i$  對應的命題等於  $D_i$ ；若  $\alpha(P_i) = Y_i$ ，且  $\beta(P_i) = D_i$ ，則  $D_i$  的信賴度等於  $Y_i$ 。

例如，定義如下：

$$FPN = \{P, T, D, I, O, f, \alpha, \beta\},$$

$$P = \{P_1, P_2\},$$

$$T = \{T_1\},$$

$$D = \{\text{it is hot, the air is dry}\},$$

$$I(T_1) = \{ P_1 \}, O(T_1) = \{ P_2 \}, f(T_1) = 0.9,$$

$$\alpha(P_1) = 0.9, \alpha(P_2) = 0,$$

$$\beta(P_1) = \text{it is hot}, \beta(P_2) = \text{the air is dry}$$

驅動前的狀況可表示如圖 3，驅動後的狀況如圖 4；

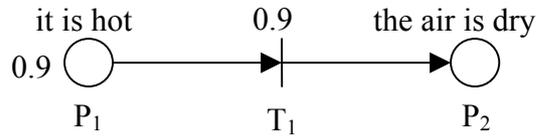


圖 3、驅動前的 Fuzzy Petri Net

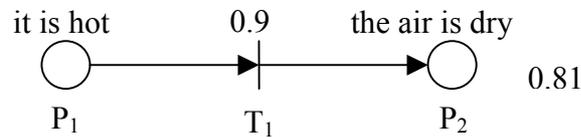


圖 4、驅動後的 Fuzzy Petri Net

而在本論文中，針對 Fuzzy Petri net 的係數作了些許的修改以符合 GQM 關係的表現：

$$FPN = (P, G, Q, M, T, D, I, O, f, \alpha, \beta),$$

其中  $T, D, I, O, f, \alpha, \beta$  的定義與上述相同，

$P = \{ p_1, p_2, \dots, p_x \}$  為有限專案規模分類位置(place)所構成的集合，

$G = \{ g_1, g_2, \dots, g_y \}$  為有限度量目標位置(place)所構成的集合，

$Q = \{ q_1, q_2, \dots, q_z \}$  為有限度量問題位置(place)所構成的集合，

$M = \{ m_1, m_2, \dots, m_n \}$  為有限度量指標位置(place)所構成的集合，

在本論文所應用的 Fuzzy Petri Net 中，除上述的 11 個元素以外，還有幾個重要的模糊係數會影響到 Fuzzy Petri Net 的分析結果，分別是每一個位置的信賴度、門檻值以及每一個 GQM 關係的可信賴參數(Certainty Factory, CF)。在本系統中，除了專案分類的信賴度是由模糊專案分類處理器計算獲得以外，度量目標、度量問題以及度量指標的信賴度，都是由其 GQM 關係的父位置之信賴度與

此 GQM 關係的可信賴參數相乘得來，而若專案分類、度量目標、度量問題或度量指標的信賴度小於門檻值，則此位置及其對應的子位置將不被輸出。以下利用一個簡單的例子來說明，如圖 5 所示：

R: IF project size is p1 THEN the g1 has be select (CF = 0.9)

其中，CF(certainty factor)表示此規則的可信賴程度。

在此例子中，我們設定門檻值為 0.6，而“project size is large”此位置的信賴度為 0.90、CF 為 0.9，因此“the goal1 will be select”此位置的信賴度為  $0.9 * 0.9 = 0.81$ ，高於門檻值 0.6，故此位置將被輸出於度量目標選單中。

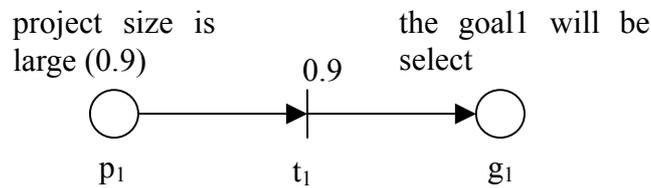


圖 5、Example of Fuzzy Petri Net

### 3.2 模糊專案分類

本系統在對知識庫要求專案規劃經驗之前，會先進行專案分類的動作，以提高經驗獲得的準確性。本論文中，依據五個專案特徵作為專案分類的依據，如表格 2 所示。其中，“使用者人數，N”，“軟體大小，S”，“開發時間，T”這三個專案特徵值，將作為“專案規模分類，P”的分類依歸。

表格 2、專案特徵值[38]

專案特徵值	細部敘述
專案參與人員數量	此軟體開發專案的參與人數
專案所開發的軟體大小	軟體預估大小(以行數為單位)
專案開發時間	專案預計的開發時間(以月為單位)
專案所開發的軟體之應用領域	例如：Web-based、嵌入式系統、資料庫系統等等
開發軟體所使用的程式語言	例如：Java、C、Visual Basic 等等

本模糊專案分類處理器的處理動作可分為兩個步驟，第一步驟為將使用者輸入的專案特徵值資料透過各自的模糊歸屬函數，轉換為模糊分類，第二步驟則將已轉換的特徵值模糊分類代入專案規模分類規則中並配合模糊推論方法進行專案規模的分類。

首先，在專案特徵值的模糊分類步驟中，由於專案規模分類需透過使用者人數、軟體大小以及開發時間這三個專案特徵值作為分類的依據，因此，當使用者輸入了使用者人數、軟體大小以及開發時間的預估值後，將會分別透過其各自的歸屬函數進行分類，在此模糊系統中，使用者人數、軟體大小以及開發時間此三個輸入值的歸屬函數如圖 6 所示。使用者只需輸入使用者人數、軟體大小以及開發時間的預估值，便可透過歸屬函數進行分析，以獲得這些專案特徵的分類及分類的信賴值，並不須以自己的主觀意識判斷這些專案特徵的分類，如此，可有效的降低使用者在進行專案特徵值分類的難度，並可避免不同使用者對特徵值分類的認知不同所造成的誤差，只需將實際的專案特徵值輸入即可透過歸屬函數獲得特徵值分類及其分類信賴值。這三個專案特徵值的分類結果將透過“Small”、“Medium”以及“Large”此三個模糊標示來表現。

例：使用者輸入  $N=15$ 、 $S=300000$ 、 $T=6$ ，如圖 7 所示，經過歸屬函數的分類之後， $N=Small(0.2)$ 、 $Medium(0.8)$ ， $S=Medium(0.8)$ ， $T=Small(0.7)$ 、 $Medium(0.3)$ 。其中每個模糊分類後的(x)，即代表此模糊分類的信賴值

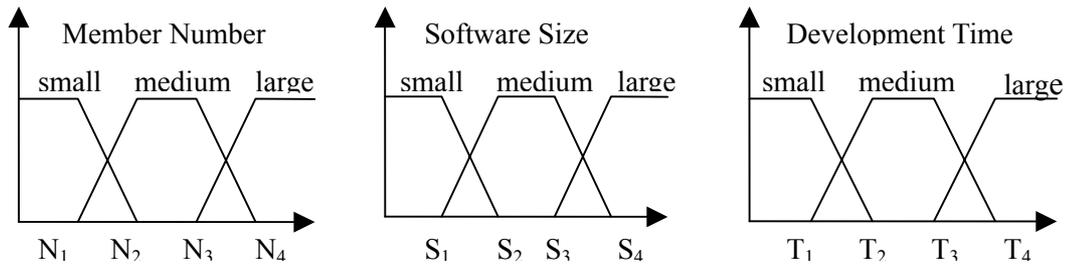


圖 6、Membership Function of N, S, and T

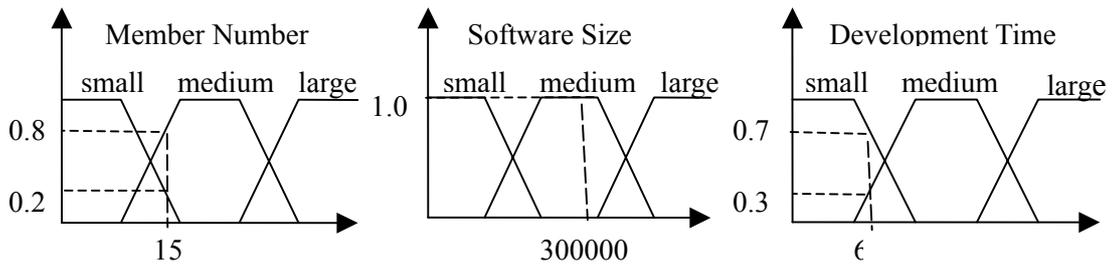


圖 7、專案特徵值分類範例

將專案特徵值模糊分類後，接下來便要進行專案規模的分類工作。專案規模分類需透過上一個步驟所產生的三個專案特徵值模糊分類的結果及其信賴值做為輸入，代入專案規模分類規則中，並配合模糊推論方法，獲得分類結果。分類的結果“專案規模分類，P”將透過四個模糊標示來表現，分別是“Small”、“Medium”、“Large”以及“Extra Large”。專案規模分類規則示意圖如圖 8 所示，例：N is Small(0.4) and S is Medium(0.6) and T is Large(0.2) Then P is Large(0.2)，此模糊規則中，P is Large 的歸屬值為 0.2 是透過模糊推論方法“min/max inference”[37]推論得知。

		N is Large				
		S	T	Small	Medium	Large
N is Medium		N is Large				
		S	T	Small	Medium	Large
N is Small		N is Large				
		S	T	Small	Medium	Large
		Small	Medium	Large	Large	Extra Large
		Small	Small	Small	Med	Extra Large
		Medium	Small	Med	Large	Extra Large
		Large	Med	Large	Extra Large	

圖 8、專案規模分類模糊規則表

本論文所採行的模糊推論方法“min/max inference”[37]之推論原則如下，第一步驟，以每條模糊規則中的最小模糊分類信賴值作為此模糊規則的信賴值，例：N is Small(0.5) and S is Large(0.9) and T is Medium(0.3) Then P is Large(0.3)。第二步驟，以對應到同樣專案規模分類的所有模糊規則中之最大信賴值作為此專案規模分類的信賴值。例：

N is Small(0.4) and S is Medium(0.6) and T is Large(0.2) Then P is Large(0.2)；

N is Small(0.5) and S is Large(0.9) and T is Medium(0.3) Then P is Large(0.3)

以上面兩條模糊規則為例，最後的推論結果，為 P is Large(0.3)

如此，便完成了專案分類的工作，接下來便可將專案分類的結果及其信賴值輸入 Fuzzy Petri Net 關係處理器中，以獲得度量規劃經驗。

## 第四章 系統架構

軟體度量指標相當相當的繁多，可度量的範圍相當廣泛，整個軟體開發流程皆可藉由許多不同的度量指標進行度量。軟體開發流程包括需求階段、分析階段、設計階段以及實作階段，每一個階段都相當的重要，且環環相扣，因此，需要許多的度量輔助工具在每個軟體開發階段輔助度量的執行，而度量規畫工具的工作便在於訂定以及調整各階段的度量工作。如圖 9 所示，度量規劃工具先針對專案的資訊需求進行分析，規劃適合的度量指標，並指派度量指標給予每個軟體開發流程階段的度量工具來進行度量工作，以此獲得軟體開發的相關資訊。

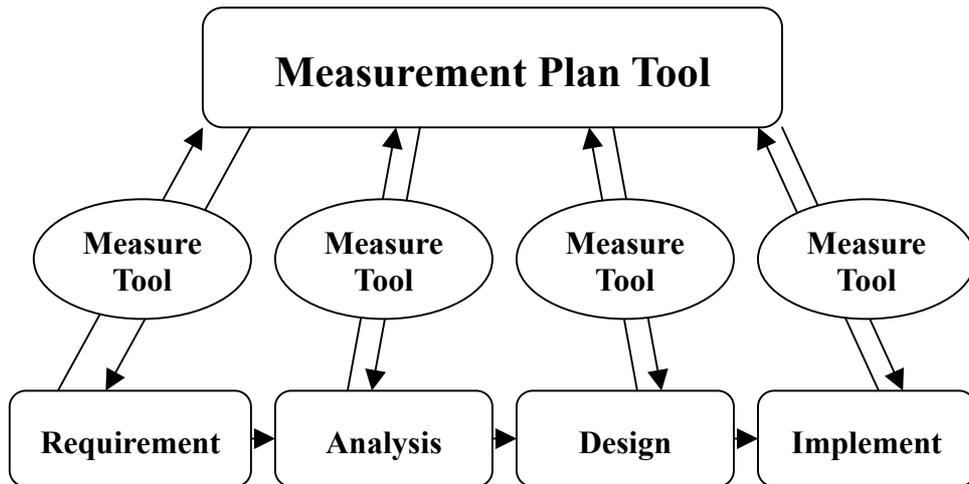


圖 9、度量工具與軟體開發流程的關係示意圖

本論文所提出的度量規劃工具最主要的目的即是輔助對度量不熟悉的專案經理進行度量的規劃。度量規劃的過程中最困難的就是選擇正確的度量指標，尤其對經驗不足的規劃人員來說。在本研究中我們將 GQM 此選擇度量指標的方法改良的更為實用，提出了一個以模糊 GQM 為基礎的度量決策支援系統(Fuzzy GQM Based Software Measurement System, FGQMSM)，本系統將目標、問題以及

度量指標三者的關係儲存於知識庫中，並以選單的方式，提供選項，並依選擇的目標選項產生對應的問題選單，最後在透過所選擇的問題選項，產出對應的度量指標。利用這樣的機制，將可避免度量規劃人員不知如何選擇度量指標的問題，並可順利將選擇的度量指標通知相關的度量工具進行後續的度量工作。系統架構圖如下圖 10 所示，使用者透過使用者介面(Graphical User Interface)與知識庫(Knowledge-Base)、GQM 處理器(GQM Processor)以及度量指標資料庫(Metric Database)進行溝通，獲得度量規劃經驗與度量指標資訊，並可透過使用者介面更新知識庫以及度量指標資料庫的內容。

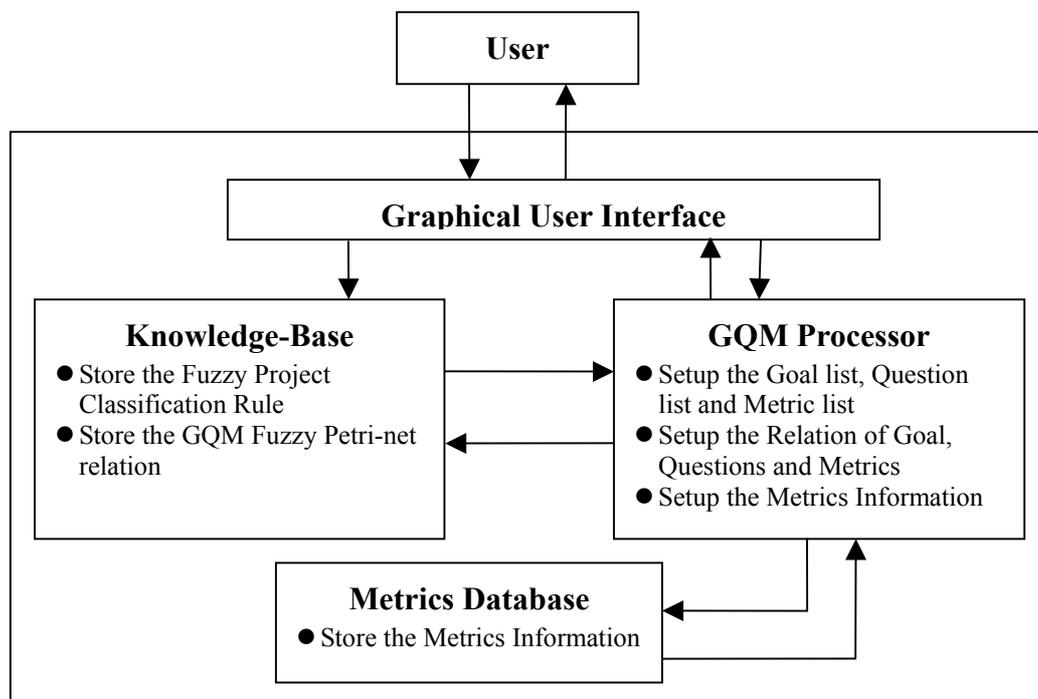


圖 10、系統架構圖

FGQMSM 總共包含了以下四個子系統：

1. 使用者介面(Graphical User Interface, GUI)：

FGQMSM 與度量規劃人員溝通的介面。FGQMSM 可透過此介面輔助度量規劃人員進行度量規劃，有經驗的度量規劃人員亦可透過此介面來輸入度量規劃經驗，使 FGQMSM 的可用性提高。

## 2. GQM 處理器(GQM Processor)：

使用者介面、知識庫以及度量指標資料庫的溝通橋樑。從知識庫中獲得的度量規劃經驗需透過 GQM 處理器處理後，產生度量目標選單給使用者介面，並針對使用者的選擇，參生對應的度量問題選單以及度量指標選單給使用者介面。而使用者選擇的度量指標同樣透過 GQM 處理器將度量指標資料庫中的資料進行處理，並產生度量指標資訊表單給使用者介面。另一方面，使用者所新輸入的度量規劃經驗與度量指標資訊，亦須經過 GQM 處理器的處理後，傳遞給知識庫與度量指標資料庫進行資料更新。

## 3. 知識庫(Knowledge-Base)：

紀錄開發專案的度量規劃經驗。包括度量目標、度量問題和度量指標的關係以及開發專案的特徵值，新的開發專案可透過專案特徵值來獲得類似專案的度量規劃經驗。

## 4. 度量指標資料庫(Metric Database)：

紀錄度量指標的詳細資訊。度量規劃工具的最終目的，便是為了選擇適用的度量指標，因此如何進行度量、度量指標的使用時機等度量指標相關資訊變顯得極為重要。

本系統最主要的兩個工作分別是軟體度量規劃以及更新系統，這兩個工作分別是由 GQM 處理器與知識庫、度量指標資料庫這三個子系統所完成的，接下來的兩節將針對這三者的關係與內部處理進行介紹。

### 4.1 知識庫與 GQM 處理器

知識庫處理的工作包括針對新專案的特徵資料進行專案模糊分類，然後從知識庫中搜尋類似的專案度量經驗，並傳遞給予 GQM 處理器。知識庫的另一項工

作為從 GQM 處理器接收新的度量經驗，並進行知識庫的更新。以下兩節便將針對這兩項工作進行介紹。

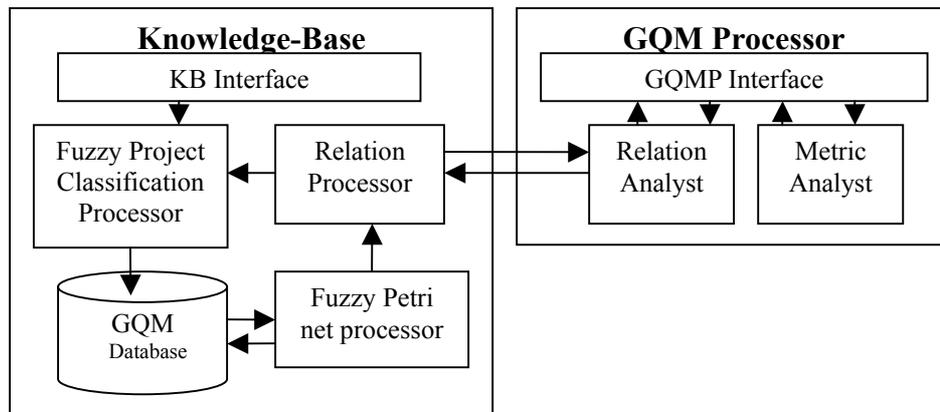


圖 11、知識庫與 GQM 處理器之內部架構與關係示意圖

#### 4.1.1 度量目標、度量問題與度量指標的關係

如圖 11 所示，首先，知識庫的知識庫介面(Knowledge-Base Interface, KB Interface)接收到度量規劃人員所輸入的專案特徵值表格 2 後，便傳遞給專案模糊分類處理器(Fuzzy Project Classification Processor)進行分析，專案的特徵值如表格 2 所示。專案模糊分類處理器將會把輸入的專案特徵值代入各自的模糊歸屬函數中進行運算，並利用 min-max inference[37]推論出此專案所隸屬的專案類型。接著便依照此專案類型至 GQM 資料庫中(GQM Database)搜尋相類似的專案度量經驗之模糊 GQM 關係，並透過 Fuzzy Petri net 處理器(Fuzzy Petri net Processor)對模糊 GQM 關係進行分析，最後將分析結果傳輸給關係處理器(Relation Processor)轉交至 GQM 處理器的關係分析器(Relation Analyst)進行下一步的處理。

GQM 資料庫中的資料結構如下圖 12 所示，專案表中儲存了不同專案的特徵值集合，而每一個專案會對應到度量目標表中的度量目標，每一個度量目標則會對應到度量問題表中的度量問題，並對應到度量指標表中的度量指標，這樣一連串的關係，我們稱之為 GQM 關係。而一組專案特徵值將會對應 GQM 儲存器

中一個以上的 GQM 關係。在本論文中，我們不只單純的紀錄專案特徵值、度量目標、度量問題與度量指標四者的關係，更加入了模糊(Fuzzy)的元素來加強其準確性與適用性。運用了模糊的觀念後，每一種專案分類與對應的度量目標之間會有一個信賴值，此信賴值會視專案分類與個別度量目標之間的關係程度高低而有所不同，因此，每一個專案分類並不一定對應相同的度量目標，需視專案模糊分類後的歸屬值。如果某一專案模糊分類的歸屬值不高，而其所對應的度量目標之信賴值也不高的話，則此度量目標被輸出的歸屬值自然不高。而這樣的模糊關係，不只存在於專案分類與度量目標之間，在度量目標、度量問題以及度量指標之間亦存在信賴值，詳細的處理流程我們將在第四章及第五章詳細敘述。因此，本論文所定義的 GQM 關係並不是單純的直接 GQM 關係，而是模糊 GQM 關係。這樣的模糊 GQM 關係在 GQM 資料庫中我們運用了 Fuzzy Petri net 的觀念來表現，並透過 Fuzzy Petri net 處理器加以分析，以獲得最適合的 GQM 關係。

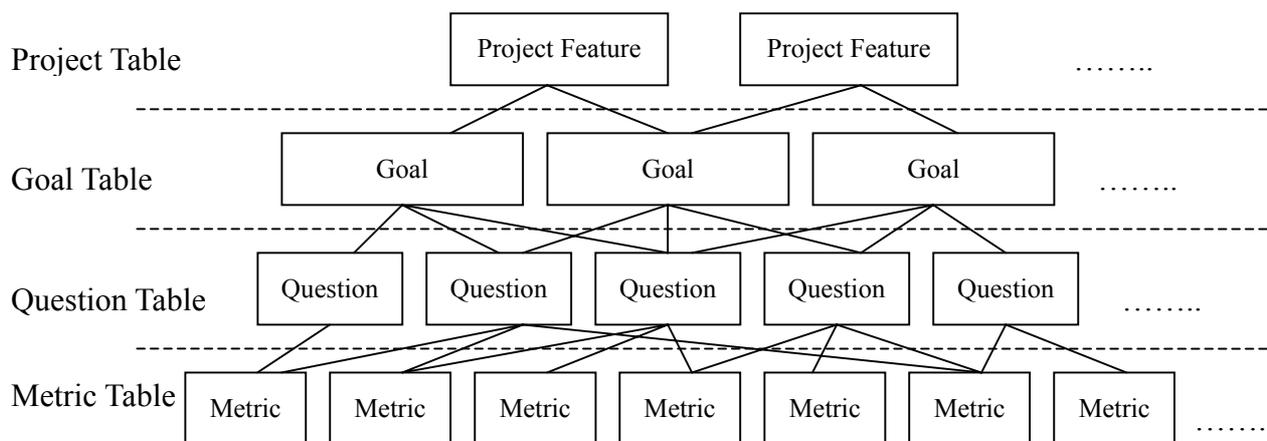


圖 12、GQM 關係資料結構示意圖

當 GQM 關係處理器(Relation Processor)接收到 Fuzzy Petri net 處理器所傳送出來的 GQM 關係後，會轉送給 GQM 關係分析器(Relation Analyst)進行處理，如下圖 13 所示，GQM 關係分析器會將接收到的 GQM 關係表格轉換為度量目標

選單、度量問題選單以及度量指標選單，最後透過 GQM 處理器介面(GQMP Interface)傳送給使用者介面(GUI)。

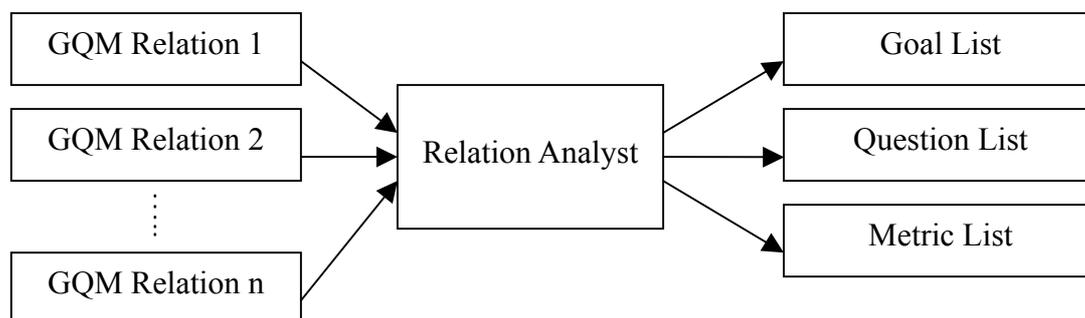


圖 13、GQM 關係分析器處理示意圖

#### 4.1.2 新增度量目標、度量問題、度量指標的關係

當 FGQMSM 針對專案特徵值所提供的度量經驗對度量規劃人員來說並不足夠時，可新增度量目標、度量問題與度量指標，並定義三者之間的模糊 GQM 關係，更新知識庫。內部的處理動作如圖 11 所示，當度量規劃人員認為度量目標選單所提供的度量目標選項並不足夠時，可直接在使用者介面新增度量目標並回傳給 GQM 處理器，當 GQM 處理器中的關係分析器接收到新增度量目標時，便通知知識庫中的關係處理器，要求知識庫中所有的度量問題與度量指標，並交由關係分析器處理為度量問題選單與度量指標選單，然後交給使用者介面呈現。度量規劃人員可從度量問題選單中選擇對應此新度量目標的度量問題，亦可新增度量問題來對應新的度量目標，但需從關係分析器所提供的度量指標選單中選擇對應此新度量問題的度量指標，並設定新的度量目標、度量問題與度量指標之間的信賴值。

新的模糊 GQM 關係將會透過 GQM 處理器的關係分析器進行分析處理，關係分析器將會把新的模糊 GQM 關係處理成 GQM 儲存器所能接受的關係表格，並傳送給關係處理器與專案特徵處理器，將 GQM 關係與專案特徵作對應，最後則把新的專案特徵值與其對應的 GQM 關係更新於 GQM 資料庫。

## 4.2 度量指標資料庫與 GQM 處理器

度量指標資料庫與 GQM 處理器之間會有兩種互動的關係：1. 度量規劃人員選擇了度量指標之後，度量資料庫將度量指標的詳細資訊傳送給 GQM 處理器處理；2. 使用者新增一個度量指標時，GQM 處理器將度量指標的詳細資訊處理後傳送給度量指標資料庫進行更新。以下兩個小節將針對這兩個工作分別介紹。

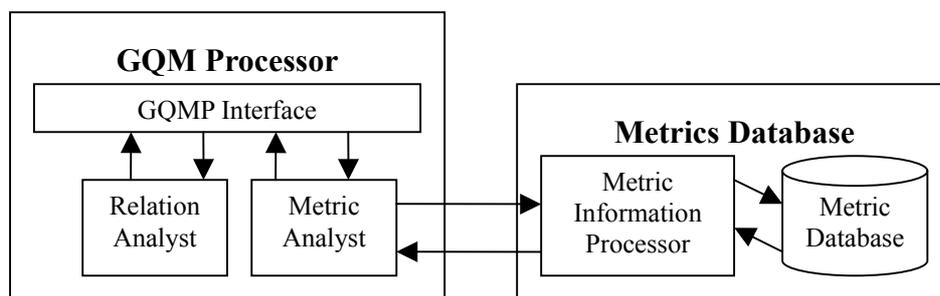


圖 14、度量指標資料庫與 GQM 處理器的關係與內部架構示意圖

### 4.2.1 傳送度量指標資訊

度量指標知識庫與 GQM 的互動關係如上圖 14 所示，度量指標 GQM 處理器透過 GQM 處理器介面(GQMP Interface)接收了度量規劃人員所選擇的度量指標名稱之後，便將度量指標的名稱傳送給度量指標分析器(Metric Analyst)，在這個處理階段，度量指標分析器並不會做分析的動作，而是直接將度量指標名稱傳送給度量資料庫的度量資訊處理器(Metric Information Processor)。度量資訊處理器在接收到度量指標名稱時，便對度量指標儲存器(Metric Storage)進行查詢的動作，並將獲得的度量指標詳細資訊傳送給度量指標分析器，度量指標分析器將會把度量指標的詳細資訊進行分析，並將度量資訊處理為易於閱讀的表單形式，最後在透過 GQM 處理器介面(GQMP Interface)傳送給使用者介面。

### 4.2.2 新增度量指標

由於新度量指標不斷的被提出，因此 FGQMSM 提供了新增度量指標的機制，以擴增 FGQMSM 的可用性。新增度量指標的系統內部處理如圖 14 所示，GQM 處理器透過接收介面接收了使用者所輸入的度量指標相關資訊後，傳送給度量指標分析器將表單中的資料擷取出來，最後在傳送給度量指標資訊處理器更新度量指標儲存器。

新增一個度量指標必須清楚定義此度量指標的相關資訊，包括的度量指標相關資訊[39]表格 3：

表格 3、度量指標相關資訊[39]

度量指標的相關資訊	細部敘述
度量指標的名稱	度量指標的簡單描述
度量指標的目的	敘述此度量指標的用途與功能
應用的方法	度量指標的使用時機以及用法
度量的公式或計算方法	度量指標的計算公式與各係數的意義
度量結果的形態	一般的度量結果為比率、序數或文字敘述
度量所需輸入的資料	資料的來源，如：程式碼、測試報告等
度量結果的需求對象	度量結果可提供給何者參考

將新增的度量指標之各項資料定義清楚後，尚須將此度量指標與知識庫中的度量問題對應清楚，如此，才算完成一個新增度量指標的工作。當然，如果此度量指標在原本的知識庫中，沒有適合的目標或問題可以對應，可為此度量指標新增度量目標以及度量問題，並更新知識庫，以此來強化 FGQMSM 的廣泛性以及適應性。

# 第五章 系統執行流程以及案例探討

在本章節中，第一部份將對本論文所提出的 FGQMSM 之系統執行流程進行介紹，再本章的第二部分中，將利用一個案例來輔助說明 FGQMSM 的執行流程。在系統執行流程方面，分為兩個方面敘述；1. 度量規劃流程：敘述一般使用者利用本系統進行度量規劃的系統執行流程；2. 更新度量知識庫流程：當度量知識庫中的資料需進行更新或調整時的系統執行流程。

## 5.1 系統執行流程

本論文所提出的 FGQMSM 以 GQM 為核心理論，利用選單的方式，克服了 GQM 在度量規劃方面實用上的困難，並提供完善的擴充機制，以保留 GQM 在度量規劃上的彈性。

### 5.1.1 度量規劃流程

本系統度量規劃的流程如圖 15 所示，在度量規劃流程中，須先輸入欲進行度量之專案的專案特徵值，FGQMSM 便會依這些專案特徵值搜尋知識庫中類似的專案之度量規劃經驗，並依此度量規劃經驗列出符合的度量目標選單。在度量目標選單中，列出的為符合此專案類型的所有度量目標選項，度量規劃人員可以輕易的從度量目標選單中選取適合的度量目標選項，以進行接下來的規劃流程。選擇度量目標選項後，便會列出對應的度量問題選項於選單中以供選擇，選擇了度量問題選項後，將會列出對應的度量指標選單，供度量規劃人員選擇適用的度量指標。每一個問題選項所對應的度量指標皆選擇完畢後，系統將會把使用者所

選擇的度量指標之詳細資料呈現給使用者，到此，整個度量規劃流程便算完成。而本系統亦會將此度量經驗更新於度量知識庫中。

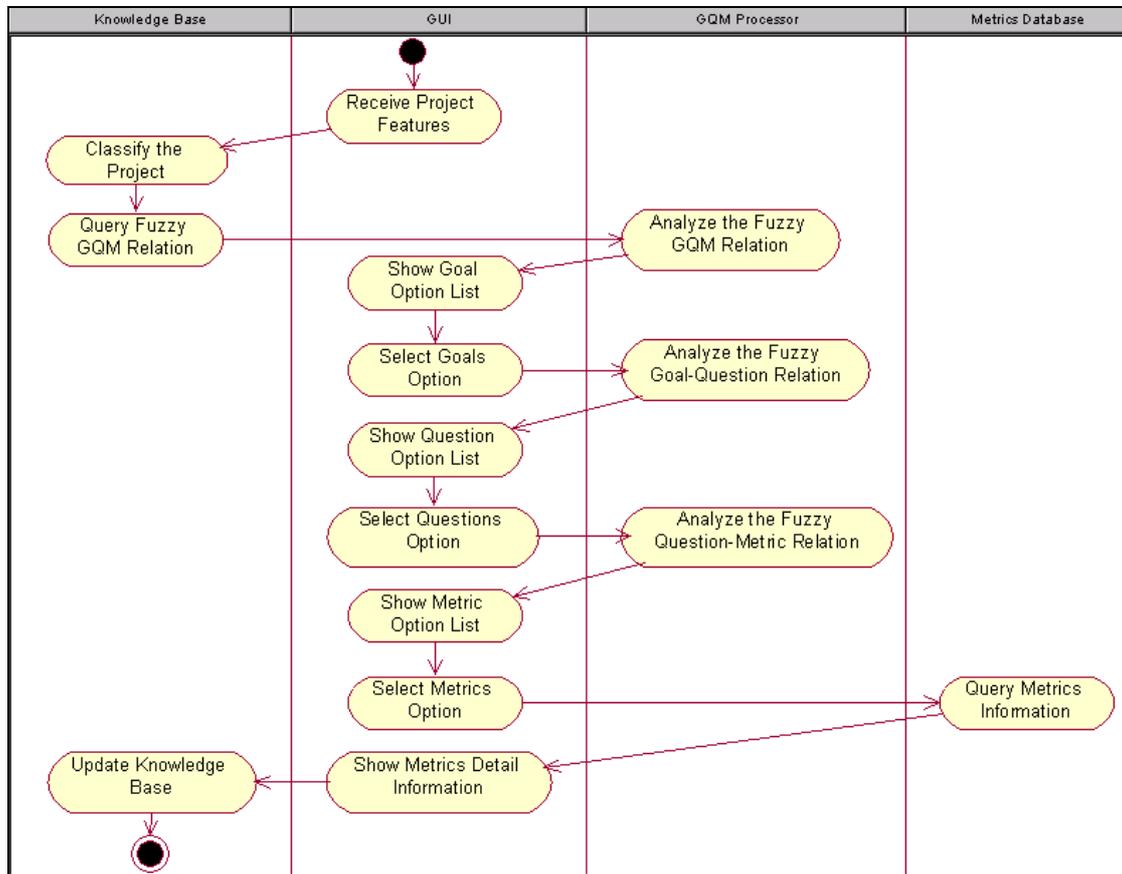


圖 15、FGQMSM 度量規劃流程圖

### 5.1.2 更新度量知識庫流程

由於時間的變遷、度量指標的增加以及度量經驗的累積等等因素，將會直接影響到度量知識庫的適用性，因此，本系統提供了完善的度量知識庫更新與調整機制。

度量知識庫更新流程如圖 16 所示，有兩個主要的更新項目，包括專案分類以及 GQM 關係。專案分類的工作由兩個元件所完成，一個是專案特徵值的歸屬函數，另一個是模糊專案分類規則。每個專案特徵值的歸屬函數將決定使用者所輸入的專案特徵值隸屬於那一個特徵值分類，而不同的特徵值分類又會直接影響

到專案的分類，因此，歸屬函數需視度量規劃人員的需求改變進行修正。而另一個與專案分類有關的元件是模糊專案分類規則，當專案特徵值經過個別的歸屬函數分析後，便需將分類的結果帶入模糊專案分類規則中，以獲得專案分類的結果。在本系統中，專案分類規則可以針對使用者的需求進行修正與調整。

另一個主要的更新項目是 GQM 關係。當度量規劃人員認為本系統中所儲存的 GQM 關係有所不足或需調整時，可透過更新 GQM 關係的機制進行處理。主要有以下幾個部分可進行更新，包括新增度量目標、度量問題或度量指標；修改度量目標、度量問題與度量指標之間的關係；以及修改度量目標、度量問題或度量指標的信賴度(certainty factor)。

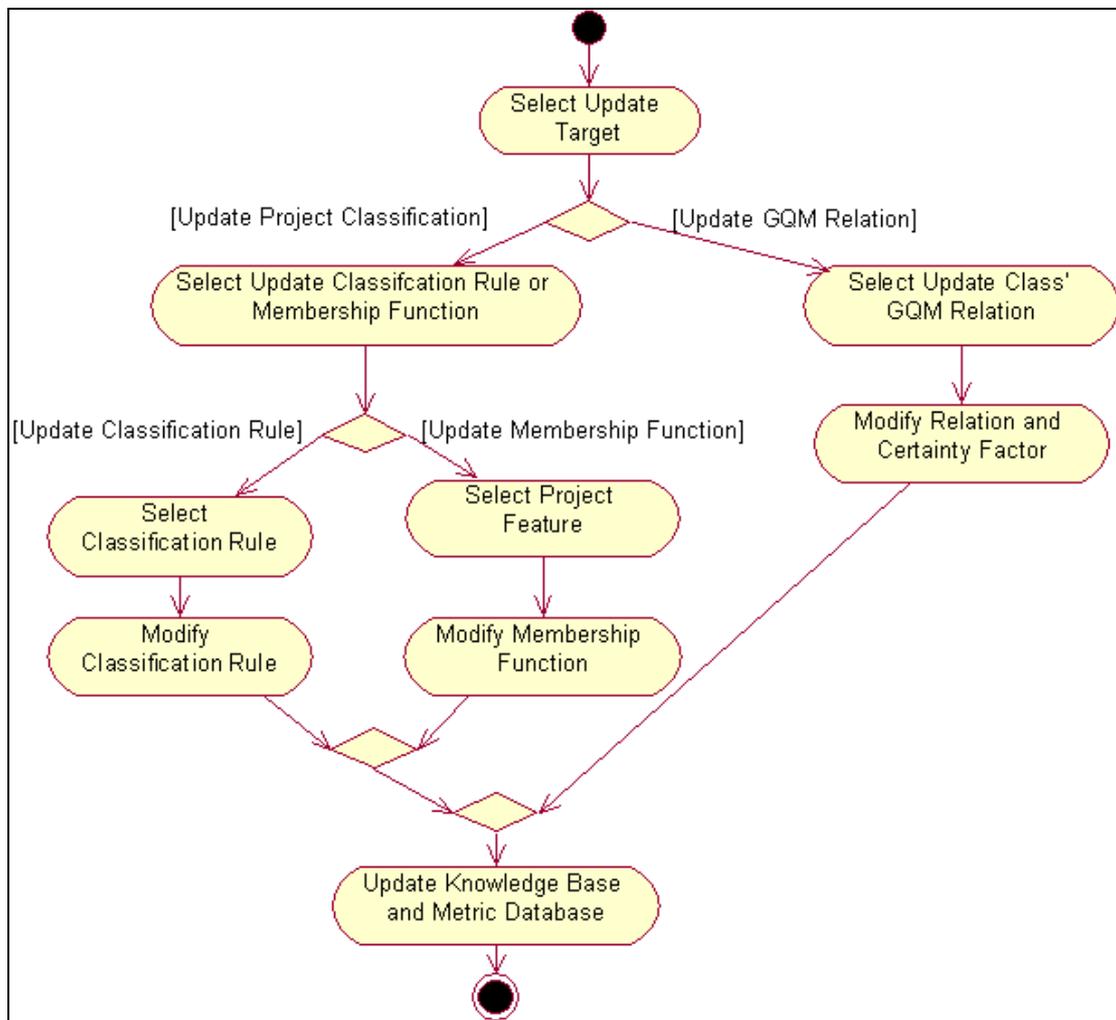


圖 16、知識庫更新流程圖

在下一小節中，本論文將利用一個案例來說明本系統進行度量規劃的流程與情形。

## 5.2 案例探討

接下來，本論文將假設一個例子來詳細描述整個 FGQMSM 的執行流程。

步驟一：輸入專案特徵值

使用者所輸入的專案特徵值如表格 4 所示。

其中，特徵值 N、S 以及 T 將透過模糊專案分類處理器進行專案規模的分類動作。

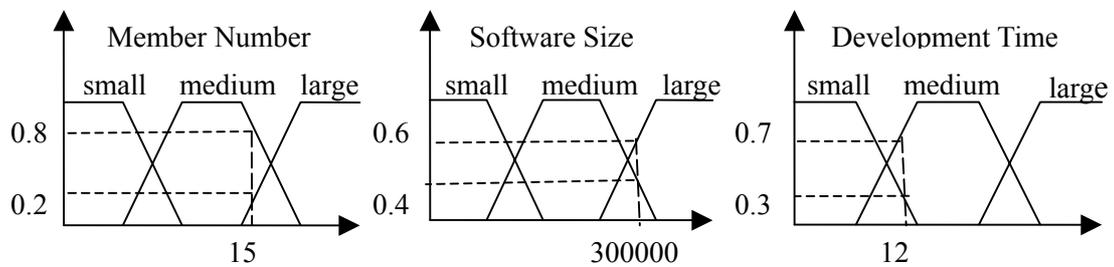


圖 17、Example of Fuzzification

步驟二：模糊專案規模分類

N、S 以及 T 是模糊專案分類處理器的三個輸入值。此處理器透過模糊化以及推論引擎來完成專案規模分類的工作。首先，此處理器使用 N、S 以及 T 的歸屬函數將使用者的輸入轉換為模糊輸入，如圖 17 所示。接下來，推論引擎將會利用專案分類規則以及模糊輸入進行推論，最後產出每一種專案規模類型的信賴度數，如表格 5 所示。

步驟三：GQM Fuzzy Petri Net

p1 表示“Large Project”專案規模類型、p2 表示“Extra Large Project”專案規模類型、g 表示度量目標、q 表示度量問題、門檻值  $\lambda = 0.40$ ，如果歸屬度數小

於門檻值則不輸出。以下為知識庫中所儲存的 GQM relation 之模糊推論規則：

- R1: IF p1 THEN g2 (CF = 0.90)
- R2: IF g2 THEN q2 (CF = 0.70)
- R3: IF g2 THEN q3 (CF = 0.85)
- R4: IF q3 THEN q5 (CF = 0.90)
- R5: IF p1 THEN g1 (CF = 0.95)
- R6: IF g1 THEN q1 and q3 (CF = 0.90)
- R7: IF p2 THEN g2 (CF = 0.85)
- R8: IF P1 THEN g3 (CF = 0.95)
- R9: IF g3 THEN q3 (CF = 0.85)

表格 4、Example of the Project Features

<b>Project Features</b>	<b>Description</b>
Member number (N)	15(people)
Software Size (S)	300000(line)
Development Time (T)	12(month)
Application Domain	Database system
Language	Java

表格 5、Example of the project classified output

<b>Project Feature</b>	<b>Description</b>
Project Size	Extra Large(0.4), Large(0.6)
Application Domain	Database system
Language	Java

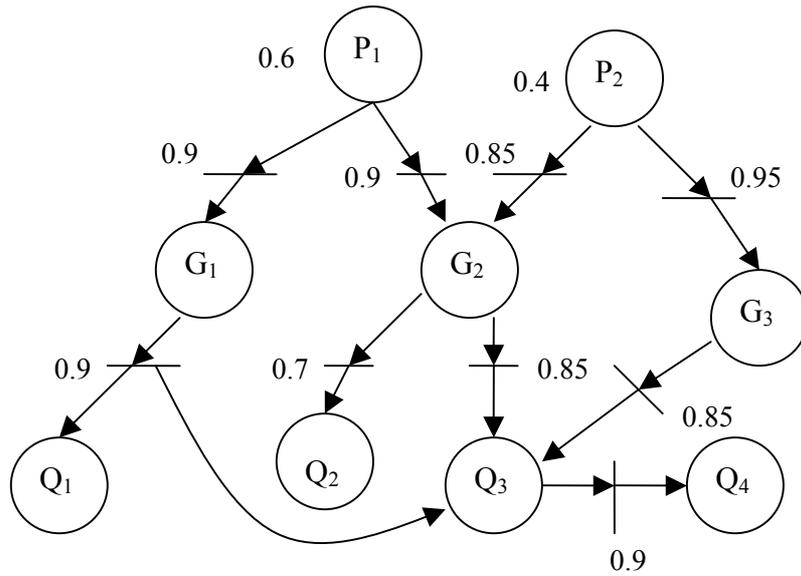


圖 18、Example of GQM Fuzzy Petri net

這些規則與參數用 Fuzzy Petri net 表示於圖 18。其中 p1 以及 p2 的歸屬度數是由模糊專案分類處理器的輸出得來。

步驟四：GQM 選單

在本例中，模糊專案分類處理器所輸出的 p1 以及 p2 之歸屬度數分別為 0.6 以及 0.2。由此可計算度量目標 {g1, g2, g3} 的信心度數：

$$g1 = 0.6 \times 0.95 = 0.57$$

$$g2 = 0.6 \times 0.9 = 0.54$$

$$g2 = 0.4 \times 0.85 = 0.34 < \lambda (0.40), \text{ cannot be fired}$$

$$g3 = 0.4 \times 0.95 = 0.38 < \lambda (0.40), \text{ cannot be fired}$$

經過計算後，g1 與 g2 的信心度數大於門檻值  $\lambda$ ，因此，將 g1 及 g2 列於度量目標選單中，交給使用者介面呈現，如表格 6 所示。

度量問題 { q1, q2, q3, q4 } 的信心度數可透過度量目標與規則的信賴度數計算獲得：

表格 6、Example of Goal list

ID	Goal
g1	Improve the timeliness of change request processing from the viewpoint of project manager
g2	Improve the reliability of the software development process in order from the viewpoint of software developer

表格 7、Example of Question list

ID	Question
q1	What is the current change request processing speed?
q3	What is the deviation of the actual change request processing time from the estimated one?
q4	Is the performance of the process improving?

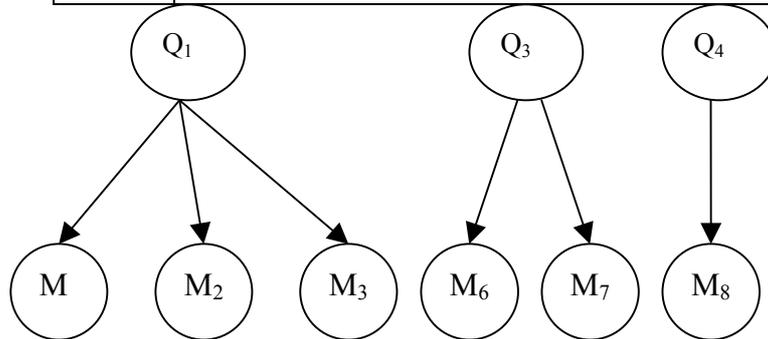


圖 19、Example of relations between Questions and Metrics

$$q1 = 0.57 \times 0.9 = 0.51$$

$$q2 = 0.54 \times 0.7 = 0.38 < \lambda (0.40), \text{ cannot be fired}$$

$$q3 = 0.57 \times 0.9 = 0.51 \text{ or } 0.54 \times 0.85 = 0.46$$

$$q4 = 0.51 \times 0.9 = 0.46 \text{ or } 0.46 \times 0.9 = 0.42$$

經過計算後，q1、q3 與 q4 的信心度數大於門檻值  $\lambda$ ，因此，將 q1、q3 與 q4 列於度量問題選單中，交給使用者介面呈現，如表格 7 所示。

前述的度量問題與所對應的度量指標之關係圖如圖 19 所示：

對應前述度量問題的度量指標將列於度量指標選單中，供使用者選擇，如表格 8 所示：

步驟五：輸出度量指標詳細資訊

表格 8、Example of Metric list

ID	Metric
m1	Response time
m2	Standard deviation
m3	% cases outside of the upper limit
m6	(Current average cycle time – Estimated average cycle time) / Current average cycle time
m7	Subjective evaluation by the project manager
m8	Current average cycle time / Baseline average cycle time

表格 9、Example of Metric Information

Metric Information	Description
Metric name	Response time
Purpose of the metric	What is the time taken to complete a specified task?
Method of application	Keep a record of each attempt
Measurement, formula and data element computations	T = ( time of gaining the result) - ( time of command entry finished )
Metric scale type	Ratio
Input to measurement	Test report and operation report
Target audience	User, Developer

表格 10、Compare fuzzily GQM relation with crisp GQM relation

	Fuzzy GQM Relation	Crisp GQM Relation
Goal listed number	2	3
Question listed number	3	4
Metrics listed number	6	8

最後，使用者所選擇的度量指標將會透過 GQM 處理器向度量指標資料庫要求詳細的度量指標資訊，並將這些詳細資訊交給使用者介面呈現給使用者。表格 9 是以度量指標 Response Time 為例子的詳細資訊。

在這個案例中，如果使用的是傳統的直接 GQM 關係而不是模糊 GQM 關係，則度量目標選單、度量問題選單以及度量指標選單將會以靜態的方式輸出給使用者，不會依照專案所歸屬的專案類型動態的調整輸出選單。應用模糊 GQM 關係與沒有使用模糊 GQM 關係的比較如表格 10。

## 第六章 系統設計與系統畫面

本章節將介紹系統在實作上的設計規劃與系統畫面。圖 20 為整個系統的 Use Case Diagram，以下將針對每個 Use Case 進行敘述，並各自以一個小節分析 Use Case 在系統內部的行為以及系統呈現的方式。

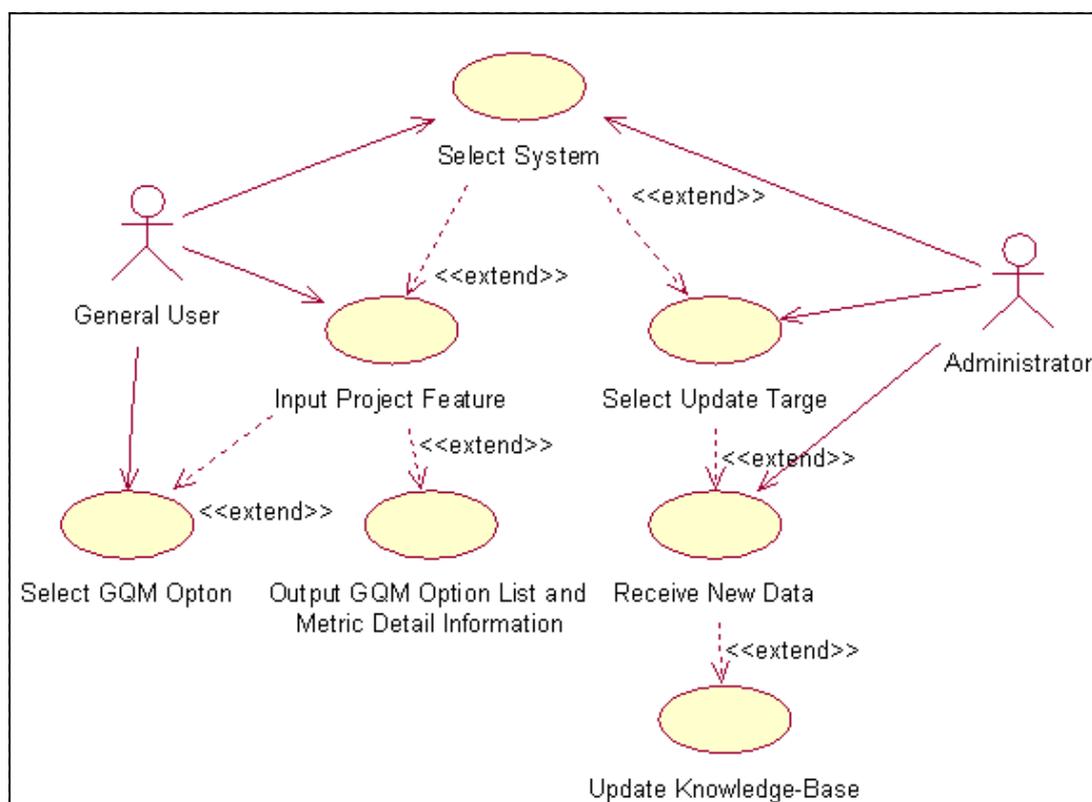


圖 20、系統 Use Case Diagram

從圖 20 中可以得知，系統的操作者共分為兩類，第一類是一般的使用者 (General user)，此類使用者利用本系統進行軟體度量規劃的工作；第二類是管理者 (Administrator)，此類使用者則是進行更新系統知識庫的工作。表格 11 中將針對所有的 Use Case 進行定義。

表格 11、Use Case 定義

Use Case 名稱	定義
Select System	由於本系統共分為兩個主要工作項目，一個為軟體度量規劃人員進行度量規劃，另一個則為軟體度量人員更新知識庫的內容，因此，在系統初始時，會先請使用者選擇此次的工作項目，而系統則會針對使用者的選擇調整顯示頁面的內容。
Input Project Feature	此為本系統進行度量規劃的第一個步驟，系統在接收到專案特徵值後，便查詢知識庫中的歸屬函數與分類規則，並進行專案分類的動作。
Output GQM Option List and Metric Detail Information	經過專案分類之後，系統便會依照分類的結果查詢知識庫，以獲得相關的模糊 GQM 關係，並立即輸出度量目標選單供度量規劃人員選擇。而度量問題選單、度量指標選單以及度量指標詳細資訊，則會依度量規劃人員的選擇，調整顯示的內容。
Select GQM Option	此 Use Case 為度量規劃人員再利用本系統進行度量規劃時，最主要的工作。當系統依照專案分類的結果輸出度量目標、度量問題及度量指標選單後，使用者需要選擇最適合的選項，便能順利的獲得度量指標及其詳細資訊，達到度量規劃的目的。
Select Update Target	在更新知識庫的工作方面，知識庫主要儲存兩類資料，第一類為模糊 GQM 關係，第二類為專案分類規則與專案徵值的歸屬函數。因此，在更新知識庫之前，需透過系統畫面中的選項進行選擇，以便系統輸出適當的資料更新頁面。
Receive New Data	當系統輸出資料更新頁面後，使用者便可依照自己的需求輸入新的資料或修改舊的資料。
Update Knowledge-Base	系統在接受到度量規劃人員輸入的新資料後，便會依照使用者選擇的更新種類，更新知識庫中的內容。此 Use Case 為系統內部的處理工作，在系統畫面的呈現方面，只會以一個簡單的提示訊息視窗，顯示知識庫更新的狀態。

接下來將依圖 20 與表格 11 中所敘述的 Use Case 分別介紹分析工作以及系統畫面：

## 6.1 Use Case: Select System

由於本系統分為兩類工作，因此在系統執行後，使用者必須先選擇欲執行的工作類型，選擇畫面如圖 21 所示。



圖 21、選擇系統畫面

而圖 22 以及圖 23 分別為選擇系統後，系統個別初始的畫面。圖 22 為度量規劃工作的初始畫面，畫面的左邊為專案特徵值的輸入位置以及專案分類的結果顯示位置，而畫面的右邊則是 GQM 選單的顯示區域。圖 23 為更新知識庫工作的初始畫面，在工具列的部分，有三個更新知識庫的選擇按鈕，而下面的區域，則為顯示知識庫資料與輸入新資料的區域。

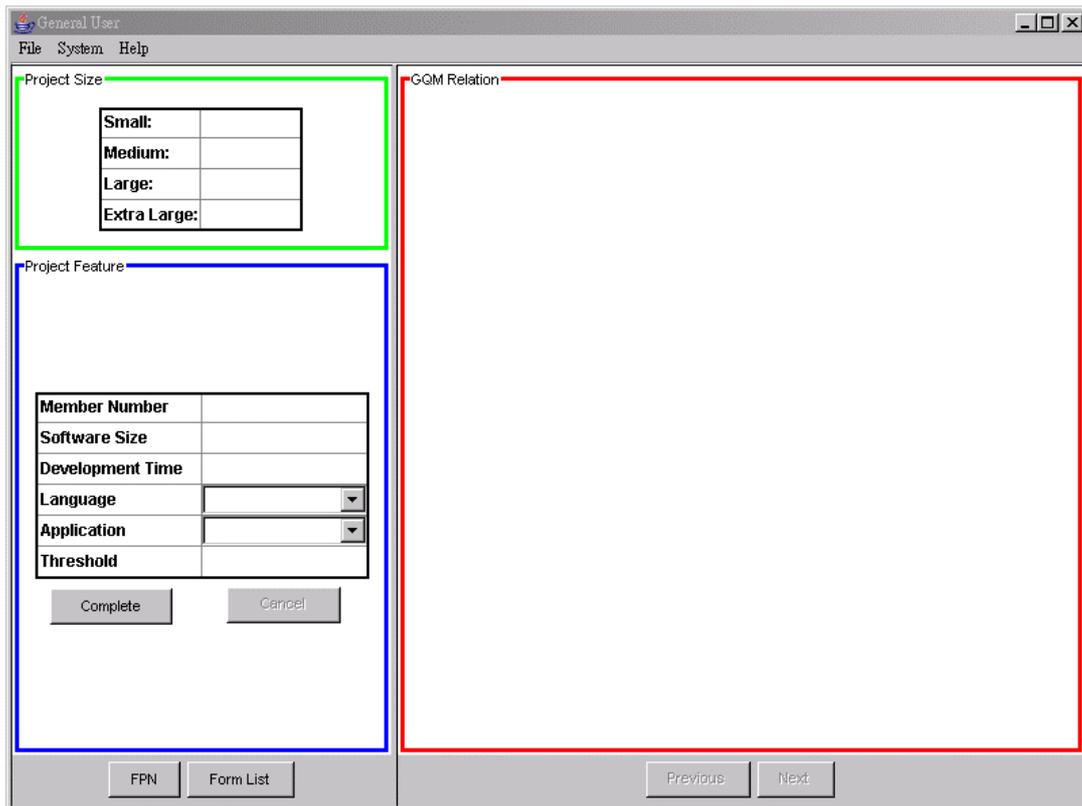


圖 22、度量規劃系統之初始畫面

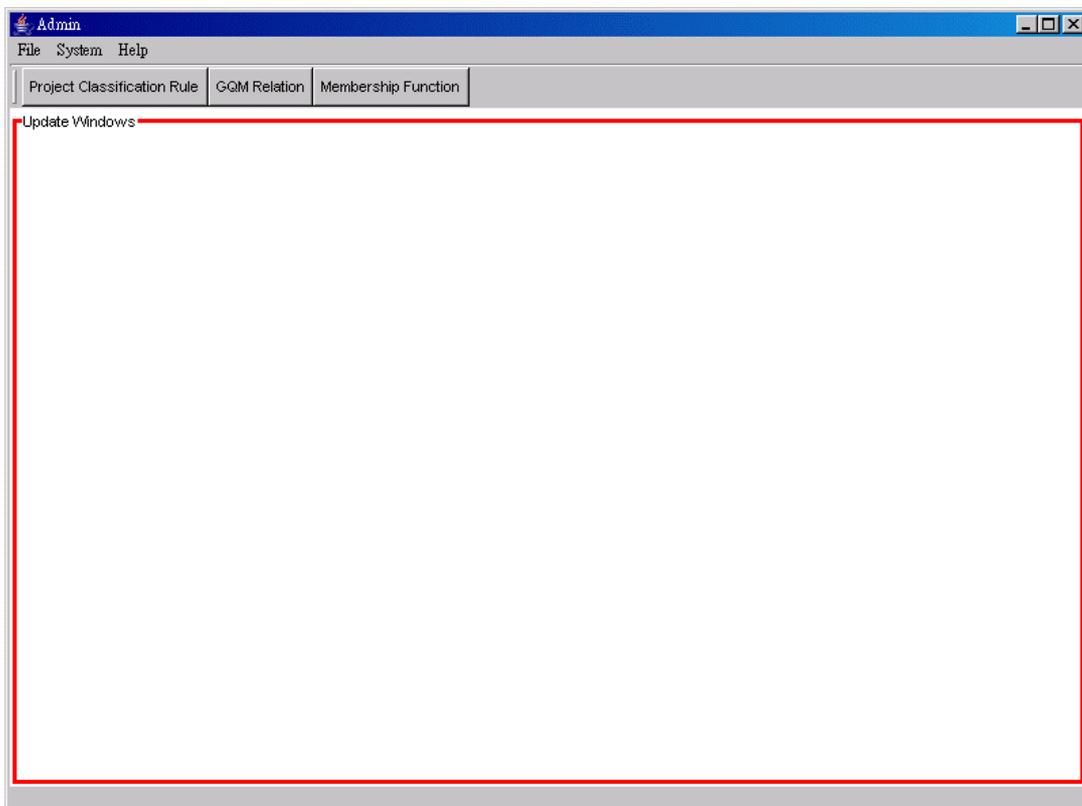


圖 23、更新系統知識庫之初始畫面

## 6.2 Use Case: Input Project Feature

本系統在進行度量規劃的第一個步驟即為輸入專案特徵值，進行專案分類。在圖 24 中可以看到五個專案特徵值的輸入位置以及專案分類的顯示位置，當使用者將專案特徵值輸入完畢後，即按下確認按鈕，系統便會依照使用者輸入的特徵值查詢知識庫當中的歸屬函數以及分類規則，即可算出此專案的專案分類並輸出於系統畫面，如圖 25 所示。

Project Size	Input
Small:	
Medium:	
Large:	
Extra Large:	

Project Feature	Value
Member Number	10
Software Size	500000
Development Time	6
Language	Java
Application	Web
Threshold	0.4

Complete      Cancel

圖 24、專案特徵值輸入位置

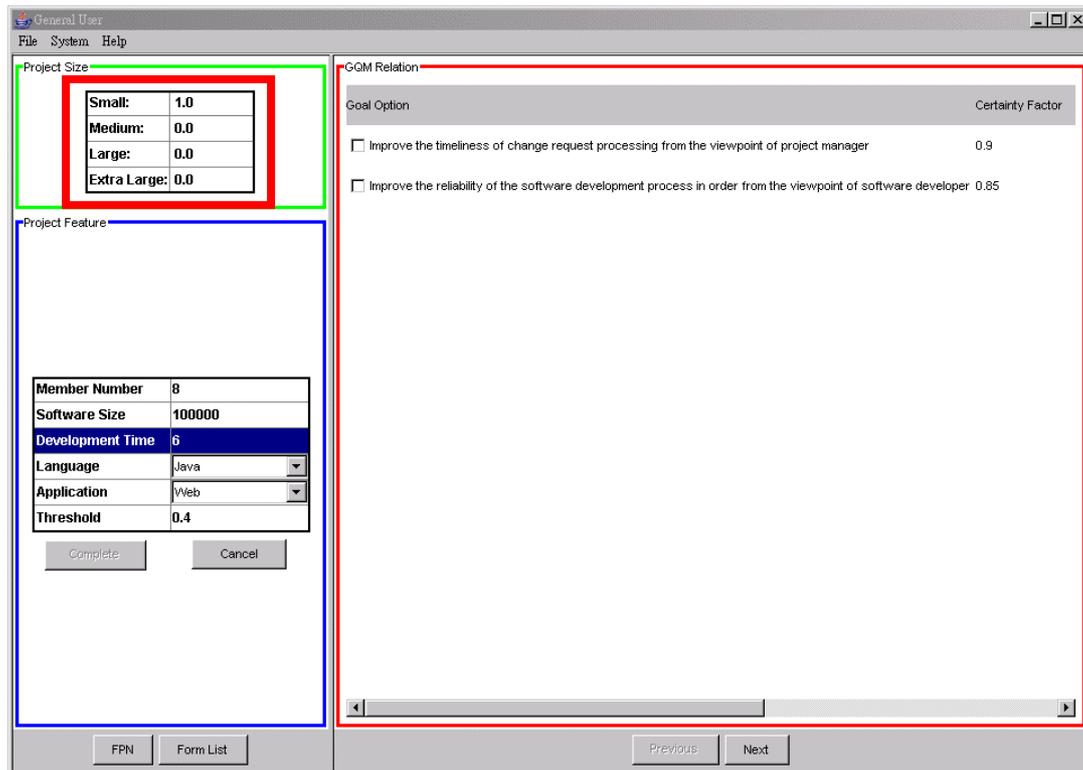


圖 25、專案分類後之結果顯示

上述的 Use Case 進行流程可以 Activity Diagram 表現，如圖 26 所示。

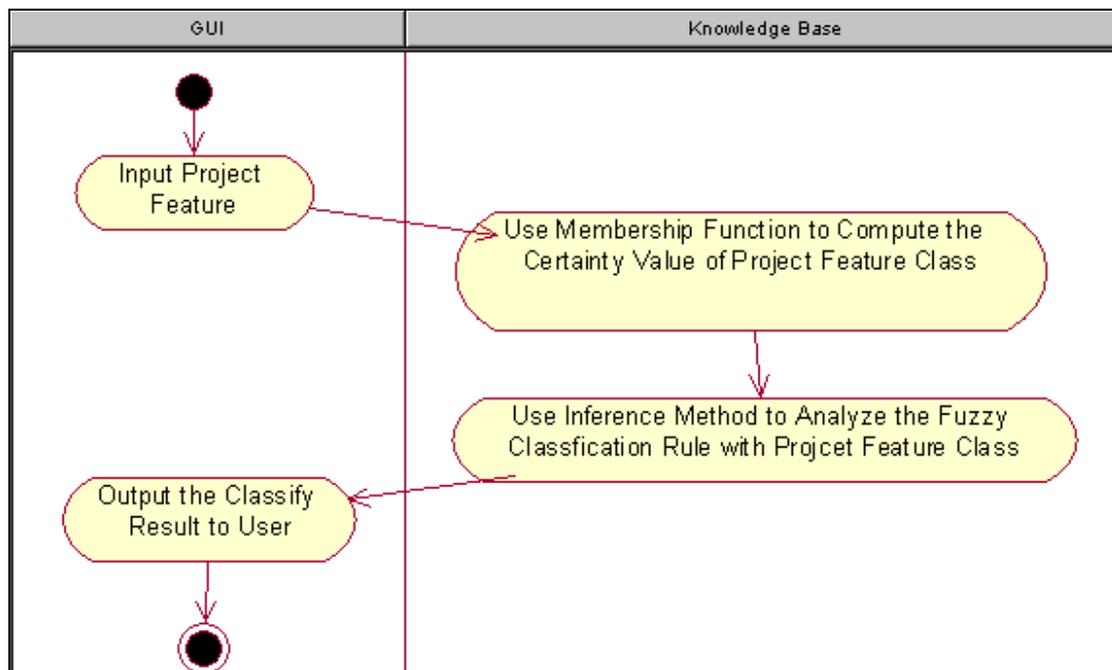


圖 26、Input Project Feature 之 Activity Diagram

## 6.3 Use Case: Output GQM Option List and Metric Detail

### Information

當系統依照使用者輸入的專案特徵值進行專案分類後，便會依專案分類的結果查詢知識庫中符合的模糊 GQM 關係，並立即將查詢到的模糊 GQM 關係轉換為度量目標選單顯示於系統畫面中，如圖 27 所示，度量目標選單顯示的項目包括點選控制項、度量目標名稱以及此度量目標的信賴度。而度量問題選單、度量指標選單以及度量指標詳細資訊的顯示格式皆與度量目標選單類似，如圖 28 及圖 29 所顯示的即為度量問題和度量指標選單，而圖 30 顯示的則為度量指標的詳細資訊，顯示的內容包含了度量指標的名稱、度量指標的目的、應用的方法、度量的公式或計算方法、度量結果的形態、度量所需輸入的資料以及度量結果的需求對象。

Project Size	Value
Small:	1.0
Medium:	0.0
Large:	0.0
Extra Large:	0.0

Project Feature	Value
Member Number	8
Software Size	100000
Development Time	6
Language	Java
Application	Web
Threshold	0.4

Goal Option	Certainty Factor
<input type="checkbox"/> Improve the timeliness of change request processing from the viewpoint of project manager	0.9
<input type="checkbox"/> Improve the reliability of the software development process in order from the viewpoint of software developer	0.85

圖 27、度量目標選單

General User  
File System Help

Project Size

Small:	1.0
Medium:	0.0
Large:	0.0
Extra Large:	0.0

Project Feature

Member Number	8
Software Size	100000
Development Time	6
Language	Java
Application	Web
Threshold	0.4

Complete Cancel

FPN Form List

GOM Relation

Question Option	Certainty Factor
<input checked="" type="checkbox"/> What is the current change request processing speed?	0.765
<input checked="" type="checkbox"/> What is the deviation of the a actual change request processing time from the estimated one?	0.675

Previous Next

圖 28、度量問題選單

General User  
File System Help

Project Size

Small:	1.0
Medium:	0.0
Large:	0.0
Extra Large:	0.0

Project Feature

Member Number	8
Software Size	100000
Development Time	6
Language	Java
Application	Web
Threshold	0.4

Complete Cancel

FPN Form List

GOM Relation

Metric Option	Certainty Factor
<input type="checkbox"/> Response time	0.765
<input type="checkbox"/> Standard deviation	0.765
<input type="checkbox"/> Subjective evaluation by the project manager	0.675
<input type="checkbox"/> %cases outside of the upper limit	0.675

Previous Next

圖 29、度量指標選單

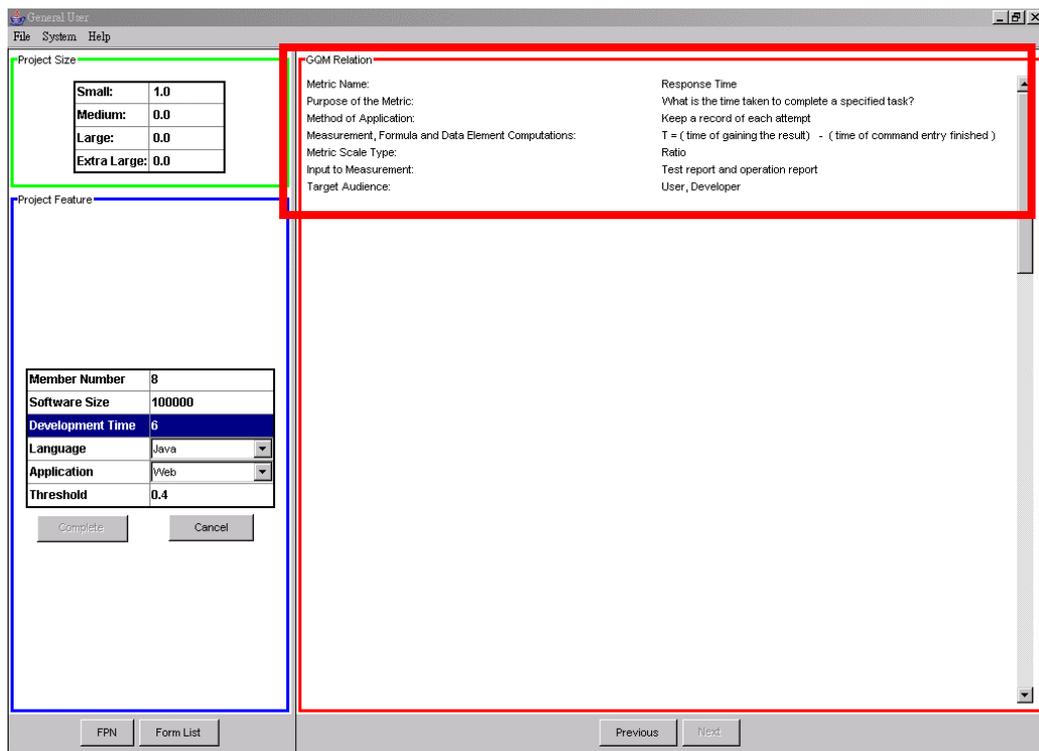


圖 30、度量指標詳細資訊列表

## 6.4 Use Case: Select GQM Option

利用本系統進行度量規劃的最終目的即是為了獲得建議採用哪些度量指標及其詳細資訊。獲得的方法即是依照系統所輸出之選單選擇適合的項目，而選擇的方式則是勾選選項前的點選控制項，並按下下一步的按鈕，系統即會依照使用者的選擇，輸出下一階段的選單。系統畫面如圖 27、圖 28、圖 29 及圖 30 所示，此四個系統畫面即是系統依照使用者的選擇，從度量目標選單一直到度量指標詳細資訊的輸出流程。

而上述的 Output GQM Option List and Metric Detail Information 以及 Select GQM Option 此兩個 Use Case 的執行流程與關係可以 Activity Diagram 表現，如圖 31 所示。

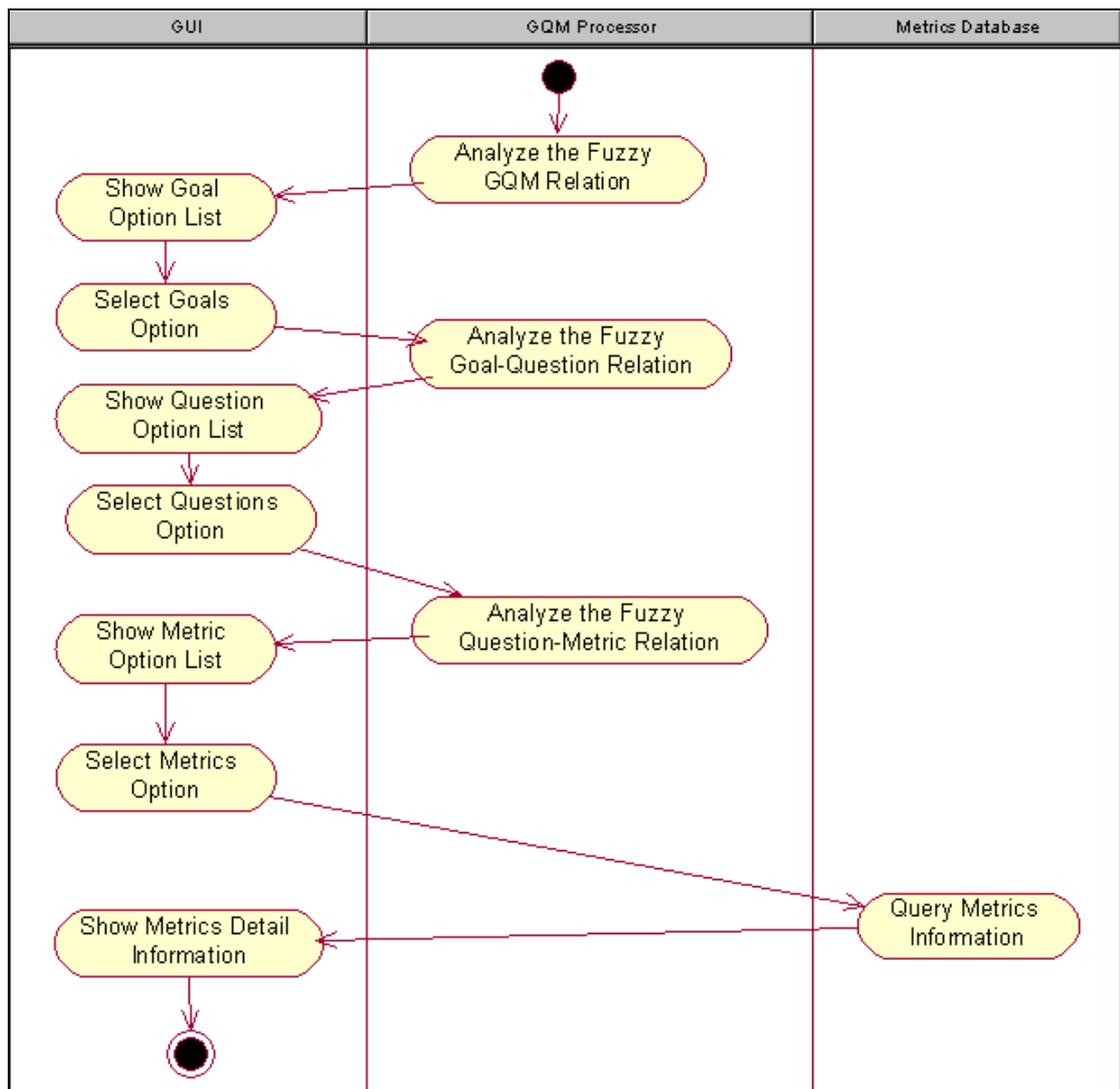


圖 31、Output GQM Option List and Metric Detail Information 以及 Select GQM Option 之 Activity Diagram

## 6.5 Use Case: Select Update Target

由於本系統中之知識庫所儲存的資料可分為三項，因此在進行知識庫資料更新的第一個步驟，便是選擇欲更新的資料項目。在圖 32 中，系統畫面的左側即為可供使用者進行更新的項目，包括模糊 GQM 關係、專案特徵值之歸屬函數以及專案分類規則。

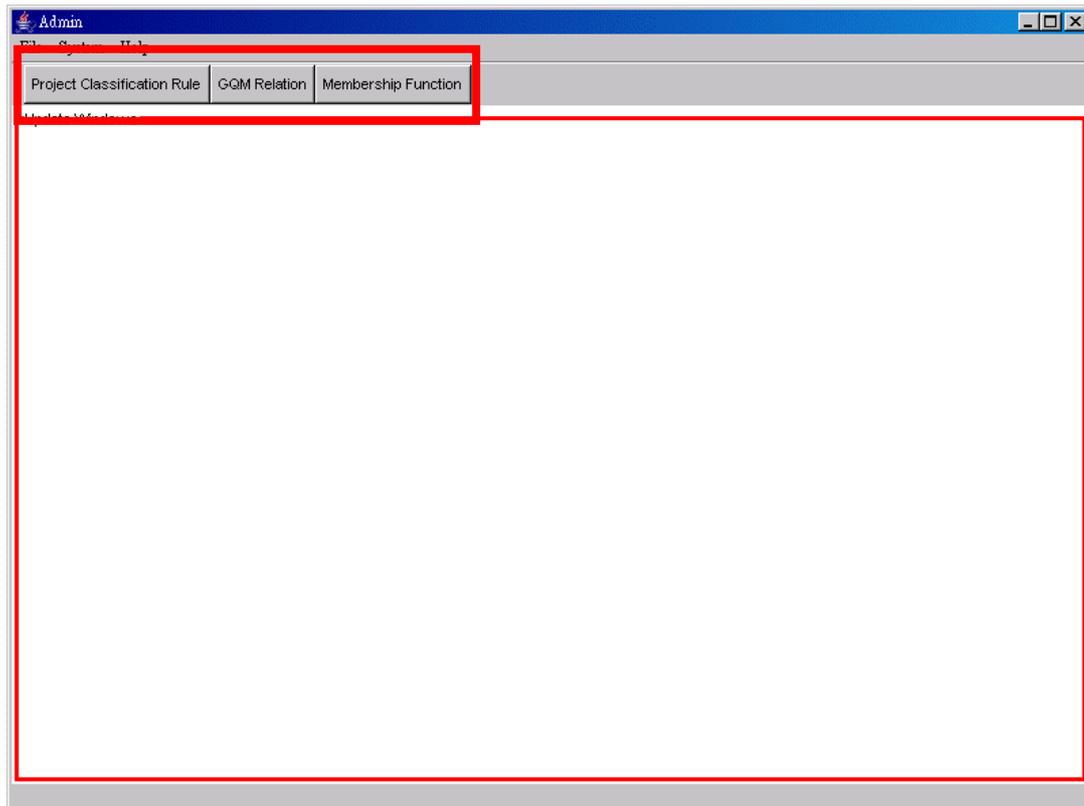


圖 32、可供選擇的知識庫更新項目

## 6.6 Use Case: Receive New Data

當系統接收到使用者選擇的更新項目後，便會依照使用者的選擇，輸出對應的資料更新輸入畫面，如圖 33 則為模糊 GQM 關係的資料更新輸入畫面，首先系統會詢問欲更新的是那一個專案類型的 GQM 關係，接著系統便會將對應於此專案類型的度量目標列出以供修改或新增，並可更新度量目標所對應的度量問題、度量指標以及度量指標詳細資訊。

圖 34 則為專案特徵值之歸屬函數的參數更新輸入畫面，度量規劃人員可依自己的需求調整轉案分類的歸屬函數，以此改變專案分類的結果。而圖 35 為專案分類規則的更新輸入畫面，透過修改專案分類規則亦可調整專案分類的結果。

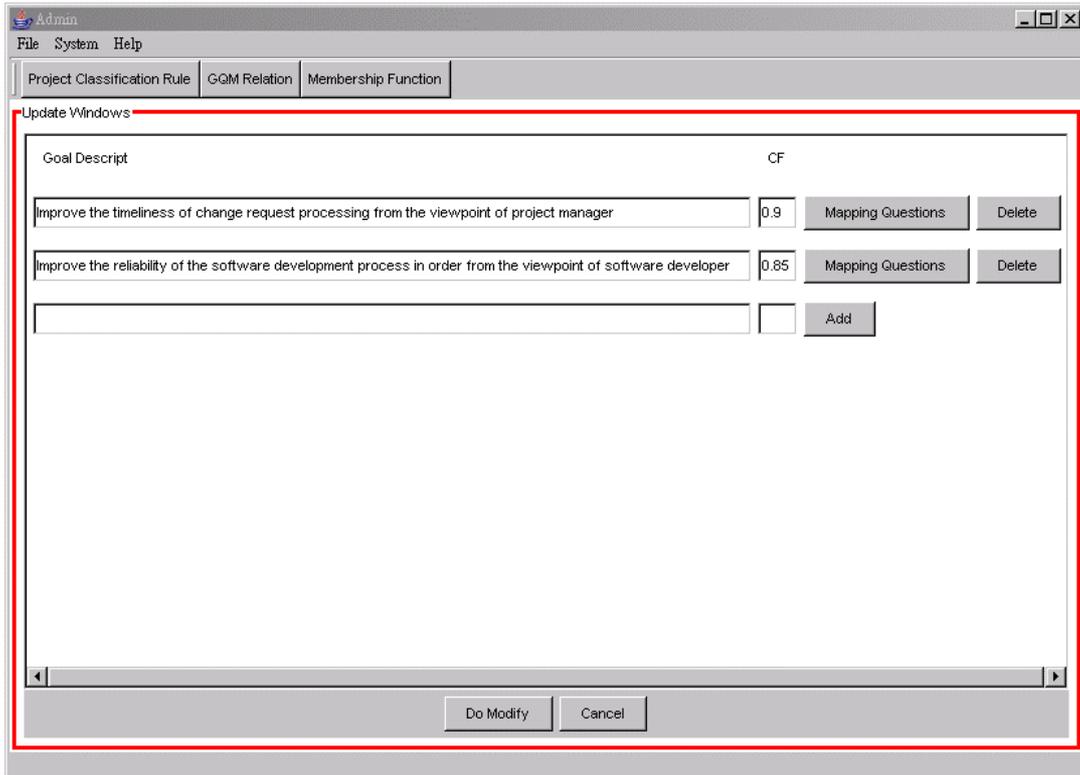


圖 33、模糊 GQM 關係的更新輸入畫面

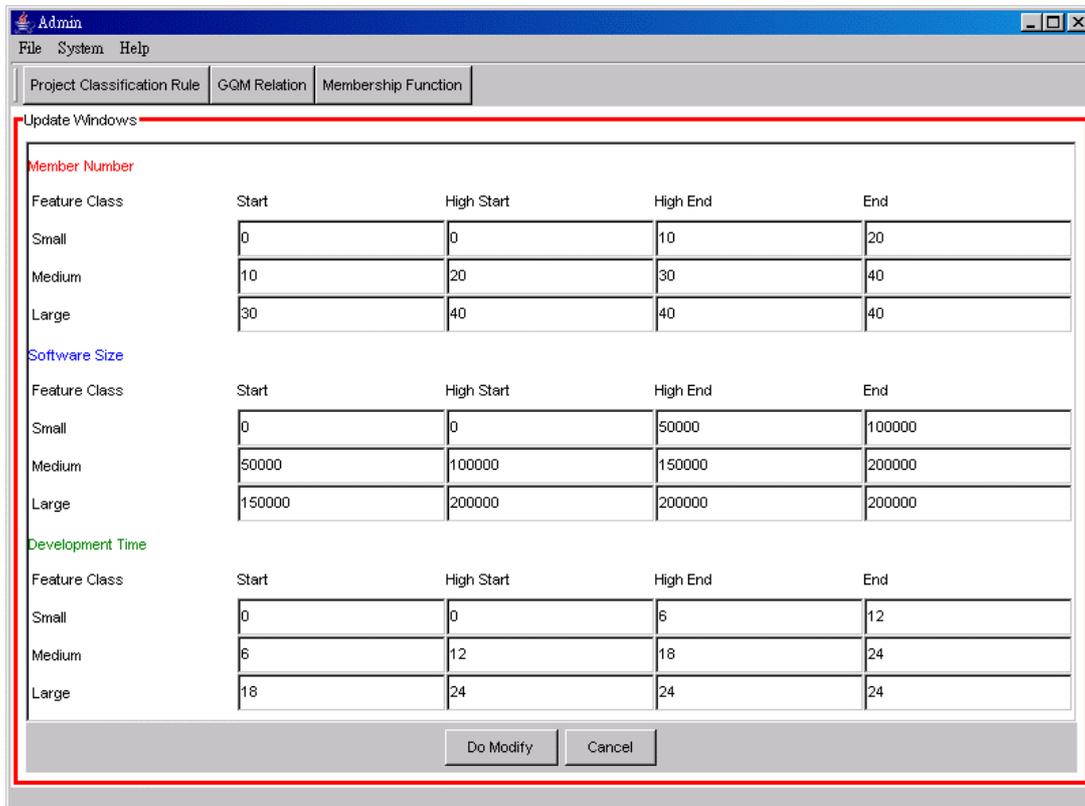


圖 34、歸屬函數的更新輸入畫面

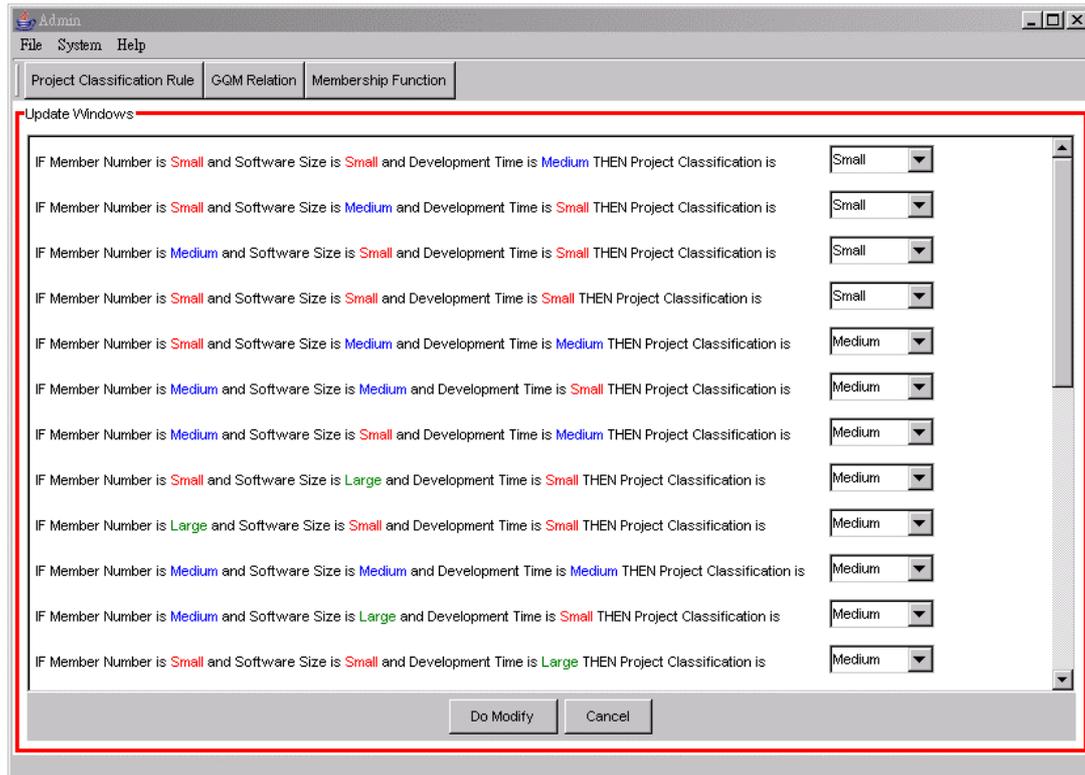


圖 35、專案分類規則的更新輸入畫面

## 6.7 Use Case: Update Knowledge-Base

當使用者將欲更新的資料輸入完成後，便可按照使用者輸入的資料更新資料庫，其執行流程可以與 Receive New Data 一起以 Activity Diagram 表現，如圖 36 所示。當系統接收到使用者的選擇後，便查詢知識庫中對應的資料，顯示於系統畫面供使用者進行資料的輸入，資料輸入完成後，知識庫便依照使用者更新的項目以及輸入的資料更新知識庫本身的資料內容。

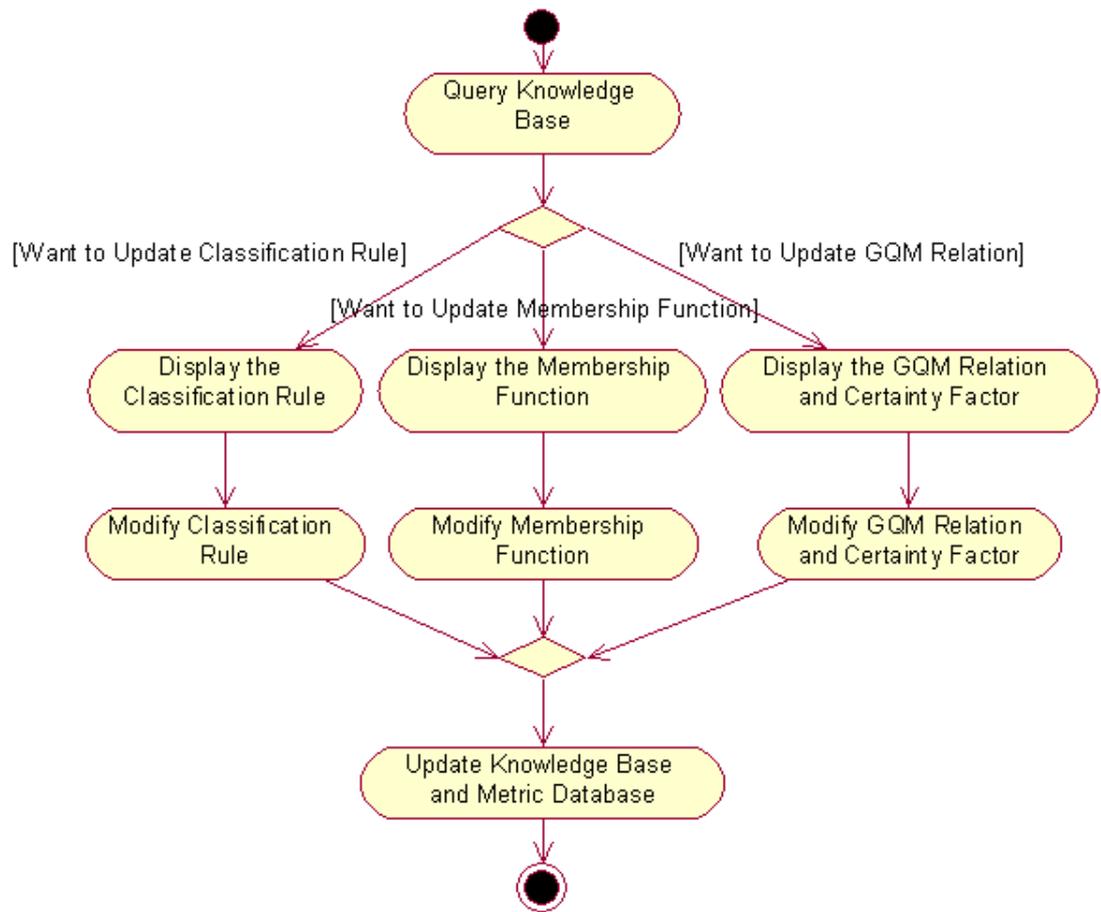


圖 36、Receive New Data 和 Update Knowledge-Base 的 Activity Diagram

## 第七章 結論與未來方向

隨著資訊時代的來臨，資訊產業的重要性日益提高，人們的生活與電腦可以說是息息相關，其中，軟體更佔了極為重要的角色。因此，軟體品質的好壞與否將直接影響我們的生活，而軟體品質也就成為目前資訊領域最重要的議題之一。軟體開發過程中可透過度量來有效控制軟體品質，但是軟體度量的導入並不容易。

在本論文中，我們提出了以 GQM 為基礎的軟體度量決策支援系統(Fuzzy GQM Based Software Measurement System, FGQMSM)，透過此系統能提高經驗較差的專案經理進行更準確且有效率的度量規劃。

FGQMSM 透過提供選單的方式幫助使用者進行度量規劃。使用者只需要從度量目標選單以及度量問題選單中選擇符合的度量目標以及度量問題，便可獲得 FGQMSM 所建議的度量指標清單以及資訊。除此之外，有經驗的專案經理亦可將自己的度量規劃經驗輸入 FGQMSM 中，使此系統可不斷的擴充。

度量規劃人員在開始進行度量規劃工作之前，必須先輸入專案的特徵值資訊，以便系統進行專案的分類，並依照此分類從知識庫中查詢適合的 GQM 關係，並產出相關的度量目標選單、度量問題選單以及度量指標選單。在專案分類以及 GQM 關係中皆加入了模糊理論的概念，專案分類方面，可令使用者能更輕易且更客觀的輸入專案特徵值，提高本系統在使用上的友善度。而在 GQM 關係方面，加入了模糊理論的元素，可提高系統輸出度量目標選單、度量問題選單以及度量指標選單的彈性，使得本系統能依照不同的專案類型，輸出更適當、更正確的度量選項。

接下來的工作，將本論文所提出之度量規劃工具，實際運用於軟體開發的過程中，以驗證本論文之想法確實能提高軟體開發的品質與效率。另外，軟體開發

各階段的度量工具，亦是我們接下來的研究重點。在進行度量規劃工具的研究中，我們發現度量工具其實是相當缺乏且不完整的，大部分的度量工具只針對某個軟體開發階段進行度量，而要整合這些不同階段的度量工具並不容易，因為他們的資料型態以及介面皆有所不同。所以，為了使得軟體度量的執行更為順利，對於軟體度量工具的研究與開發將是未來的研究重點。

## 第八章 參考文獻

- [1] Carnegie Mellon University, Software Engineering Institute.  
<http://www.sei.cmu.edu/cmm/cmms/cmms.html>
- [2] T. V. D. Wiele and A. Brown, “ISO 9000 Series Certification Over Time: What Have We Learnt?”
- [3] V. Basili, “Software Quality Assurance and Measurement: A Worldwide perspective, Applying the Goal/Question/Metric Paradigm in the Experience factory,” *International Thomson Computer Press, ITP An International Thomson Publishing Company*, Chapter 2, pp. 21- 44, 1995.
- [4] Y. Wang and Q. He, “A Practical Methodology for Measurement Deployment in GQM,” *Proceedings of the 2003 Canadian Conference on Electrical and Computer Engineering (CCECE '03)*, Vol. 2, pp. 1329-1332, May 4-7 2003.
- [5] A. Fuggetta, L. Lavazza, S. Morasca, S. Cinti, G. Oldano and E. Orazi, “Applying GQM in Industrial Software Factory,” *ACM Transactions on Software Engineering and Methodology*, Vol. 7, No. 4, pp. 411-448, October 1998.
- [6] D. S. Yeung and E. C. C. Tsang, “Fuzzy Knowledge Representation and Reasoning Using Petri Nets,” *Expert System Application*, vol. 7, pp. 281-290, 1994.
- [7] B. R. Gaines and M. L. Shaw, “From fuzzy logic to expert systems,” *Information Sciences*, vol. 36, pp. 5-15, 1985.
- [8] --, *Software Engineering – Software Measurement Process Framework*, 1999, ISO/IEC WD 15939.
- [9] --, *The Capability Maturity Model – Guidelines for Improving the Software*

- Process. Software Engineering Institute, Carnegie Mellon University: Addison Wesley Longman Inc., 1994.
- [10] T. Hall and N. Fenton, "Implementing Software Metrics – the Critical Success Factors," *Software Quality Journal*, vol. 3(4), pp. 195-208, 1994.
- [11] M. Berry and R. Jeffery, "An Instrument for Assessing Software Measurement Programs," *Empirical Software Engineering An International Journal*, vol. 5(3), pp. 183-200, 2000.
- [12] Alfonso Fuggetta, Luigi Lavazza, Sandro Morasca, Stefano Cinti, Giandomenico Oldano, Elena Orazi, "Applying GQM in an Industrial Software Factory," *ACM Transactions on Software Engineering and Methodology*, vol. 7(4), pp. 411-448, 1998.
- [13] Y. Mashiko and V. Basili, "Using the GQM Paradigm to Investigate Influential Factors for Software Process Improvement," *In Journal of Systems and Software*, vol. 36, pp. 17-32, 1997.
- [14] I. S. Bhandari, M. G. Mendonça, and J. Dawson, "On the Use of Machine-Assisted Knowledge Discovery to Analyze and Reengineer Measurement Frameworks," *Proceedings of CASCON'95*, Toronto, Canada, pp. 275-284.
- [15] S. G. MacDonell, "Deriving Relevant Functional Measures for Automated Development Project," *Information and Software Technology*, vol. 35(9), pp. 499-512, 1993.
- [16] Gray, S.G. MacDonell, "GQM++ A Full Life Cycle Framework for the Development and Implementation of Software Metric Programs," *In Proceedings of ACOSM'97 Fourth Australian Conference on Software Metrics*, Canberra, Australia, ASMA, pp. 22-35, 1997.
- [17] T. Olsson, P. Runeson, "V-GQM: A Feed-Back Approach to Validation of a

- GQM Study,” *Software Metrics Symposium, Proceedings Seventh International*, 4-6 April 2001.
- [18] F. Niessink, H. van Vliet , “Measurements Shoud Generate Value, Rather Than Data,” *6th International Software Metrics Symposium*, pp. 31-38, 1999.
- [19] A. Birk, R. van Solingen, J. Jarvinen, “Business Impact, Benefit, and Cost of Applying GQM in Industry: An Indepth, Long-Term Investigation at Schlumberger RPS,” *5th International Software Metrics Symposium*, pp. 93-96, 1998.
- [20] R. Offen, R. Jeffery, “Establishing Software Measurement Programs,” *IEEE Software*, 14(2), pp. 45-53, 1997.
- [21] L. A. Zadeh, “Fuzzy Sets,” *Information and Control*, vol. 8, pp. 338-353, 1965.
- [22] K. S. Leung and W. Lam, “Fuzzy Concepts in Expert Systems,” *IEEE Computer*, pp. 43-56, September 1988.
- [23] T. Lotan, H. N. Koutsopoulos, “Models for Route Choice Behavior in the Presence of Information Using Concepts from Fuzzy Set Theory and Approximate Reasoning,” *Transportation 20*, pp. 129-155, 1993.
- [24] X. Li and W. Yu, “Object Oriented Fuzzy Petri Net for Complex Knowledge System Modeling,” *IEEE Conference on Control Applications*, pp. 476-481, 2001.
- [25] R. L. de Mantaras and L. Godo, “From fuzzy logic to fuzzy truth-valued logic for expert systems: a survey,” *Second IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 750-755, 1993.
- [26] S. M. Chen, “A fuzzy reasoning approach for rule-based systems based on fuzzy logics,” *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 26, no. 5, pp. 769-778, October 1996.
- [27] S. M. Chen, “Knowledge Representation Using Fuzzy Petri Nets,” *IEEE*

- Transactions Knowledge and Data Engineering*, vol. 2, no.3, pp. 311-319, Sept. 1990.
- [28] J. F. Peters, A. Skowron, Z. Surai, W. Pedrycz, and S. Ramanna, "Approximate Real-Time Decision Making : Concepts and Rough Fuzzy Petri Net Models," *International Journal of Intelligent Systems*, vol. 14, no. 8, pp. 805-839, Aug. 1999.
- [29] Y. Zhou and T. Murata, "Modeling and Analysis of Distributed Multimedia Synchronization by Extended Fuzzy-Timing Petri Nets," *Journal of Integrated Design and Process Science*, Vol. 4, No. 4, pp. 23-38, December 2001.
- [30] S. M. Chen, "Weighted fuzzy reasoning using weighted fuzzy Petri nets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 386-397, March/April 2002.
- [31] A. Knoar and A. K. Mandal, "Uncertainty Management in Expert Systems Using Fuzzy Petri Nets," *IEEE Transaction on Knowledge and Data Engineering*, vol. 8, no. 1, pp. 96-105, Feb. 1996.
- [32] C. C. Looney, "Fuzzy Petri Nets for Rule-Based Decisionmaking," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 178-183, Jan./Feb. 1988.
- [33] H. Sacrpelli, F. Gomide, and R. R. Yager, "A Reasoning Algorithm for High Level Fuzzy Petri Nets," *IEEE Transaction on Fuzzy Systems*, vol. 4, no. 3, pp. 282-294, Aug. 1996.
- [34] D. S. Yeung and E. C. C. Tsang, "A Multilevel Weighted Fuzzy Reasoning Algorithm for Expert Systems," *IEEE Transaction on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 28, no. 2, pp. 149-158, Mar. 1998.
- [35] P. Carinena, A. Bugarin, S. Fraga, and S. Barro, "Enhanced Fuzzy Temporal

- Rules and Their Projection onto Fuzzy Petri Nets,” *International Journal of Intelligent Systems*, vol. 14, no. 8, pp. 775-804, Aug. 1999.
- [36] F. Wu, “Fuzzy Time Semirings and Fuzzy-Timing Colored Petri Nets,” *International Journal of Intelligent Systems*, vol. 15, no. 6, pp. 812-818, Nov./Dec. 1985.
- [37] C. Lee, “Fuzzy Logic in Control Systems: Fuzzy Logic Controller— Part I, II,” *IEEE Transactions on Systems Man and Cybernetics*, vol. 20(2), pp. 404-435, 1990.
- [38] 軟體開發技術文件手冊，經濟部工業局，1992年。
- [39] ISO/IEC 9126, International Standard, Information Technology – Software Product Evaluation – Quality characteristics and guidelines for their use, 1991.