

以本體論的觀點建構軟體專案管理的知識庫系統

—以國內某科技大學軟體專案研究中心為實證分析對象

黃明祥

屏東科技大學資管研究所 副教授
mshuang@mail.npust.edu.tw

林瑋堉

屏東科技大學資管研究所 研究生
m9756037@mail.npust.edu.tw

摘要

近年來，本體論廣泛應用在醫療、稅務以及圖書館等，基於軟體專案管理之知識領域涵蓋範圍內容十分廣泛，本研究擬採用業界應用最為廣泛之三個軟體開發作業程序標準：CMMI、PMBOK、IEEE，應用在一個軟體專案開發的流程，包括系統需求、系統規劃、系統需求分析、系統設計、程式設計、系統測試以及系統部署與結案等階段。其中軟體專案知識庫系統涵蓋應用知識領域、知識本體、專案技術、專案文件、專案案例等四大類型知識。本體論具能夠描述特定知識領域內先關概念以及彼此關係鏈結，因此，本研究以本體論發展出一個適用於軟體專案領域知識的分類方法，提出以本體論為基礎的內容分類檢索，分類機制較關鍵詞分類好的檢索機制，利用本體論來表達軟體專案領域知識，協助系統瞭解使用者查詢的語意，提供具有語意的查詢服務，從而獲得豐富而完整的相關資訊，再依據上述分類法建構出以本體論為基礎的專案管理知識庫系統，不但提供較關鍵詞比對更好的檢索機制，更提供分散式軟體開發團隊的環境中使用，協助進行知識管理活動，將軟體知識進行有效的擷取、組織、精煉與分佈，以支援各階段軟體開發工作進行，落地實知識分享之安全性控管。

關鍵詞：軟體專案管理、知識分享、知識庫系統、本體論。

1. 前言

在智慧型系統的研究領域中，許多研究在發展知識庫幫助使用者從資訊中取得所想要得到的知識，而收集及整理資訊並把資訊快速的傳送給需要的人，持續分析以維護資訊有效性的過程與機制是必要的關鍵，又以高度知識密集的軟體專案更是迫切需要，所以，本研究利用人工智慧所應用的本體論 (Ontology) 來充分建構人類的思維基礎，模擬人類對專案知識分類的知識概念，建構一軟體專案知識庫系統，將專案知識分類的知識外顯化，提供專案人員搜尋專案知識的知識庫系統，除了提供專案人員正確的知識外，亦可做為學校訓練及實務界在軟體專案開發之參考。

本研究以軟體開發人角度來瞭解執行軟體開發作業時之需求，設計一個知識本體 (Knowledge-ontology)，整合開發流程 (Capability Maturity Model Integrated, CMMI)、PMBOK (Project Management Body of Knowledge) 與 IEEE (Institute of Electrical and Electronics Engineers) 三大標準、專案文件、專案技術、專案參考案例、流程技巧的整合性網站。在研發知識庫系統時，考慮重點為如何使專案人員能快速有效擷取所需資訊，以及如何利用網路服務技術快速蒐集軟體專案的知識，並建置知識庫在專案開發作業中運用，以加速軟體專案的開發效率、正確性及知識的傳承，達成提昇知識分享效率的最終目的。依據本體論之樹狀架構發展出知識分類，並以此為基礎建構知識庫系統，進而更快速達到搜尋效率。根據 Mike Uschold 等人對本體論的定義，「本體論是一個大家共享

的概念」(Uschold et al., 1996)，它可應用在知識分類之中，利用本體論技術，讓系統能了解使用者真正的語意，以智慧型推理機制，輔助軟體專案開發正確性，提供軟體專案團隊在知識分享的輔助工具並支援軟體開發流程之進行，有助於提升知識的分享與再用性，進而作為實務界建立與應用相關知識庫系統之參考依據。

軟體專案開發的環境設計為針對軟體專案團隊發展一套以本體論為基礎的軟體專案知識庫系統，不僅僅是探討知識分享應用在軟體專案開發之重要議題、相關理論，更進而實作出提供軟體專案團隊在知識分享的輔助工具並支援軟體開發流程之進行，以及不斷地累積軟體專案開發過程之經驗與技術，提升知識的分享與再用性，作為實務界建立與應用相關知識庫系統之參考依據。本研究目的敘述如下：(一)探討以本體論的理論基礎建構一套軟體專案知識庫系統之重要議題，(二)探討本體論如何能提升知識庫系統功能及建構時重要的因素，與(三)藉由知識庫系統探討軟體專案管理知識庫系統應用在專案團隊知識分享活動的績效。

2. 文獻探討

本研究之重點在於發展一個以本體論為基礎的軟體專案管理知識庫系統並透過發展的知識庫系統來協助專案團隊進行知識分享並支援各階段軟體開發活動。因此，本研究相關的文獻主要有：(一)軟體專案的開發流程，(二)軟體專案的知識管理，(三)知識本體論的應用，(四)軟體專案知識庫的開發。茲將各相關文獻分述整理如下：

2.1 軟體專案的開發流程

軟體開發程序是資訊技術的核心，如何提高軟體的品質始終是計算機科學領域研究的重要方向之一(張紅延，2004)。適當的軟體開發管理方法與知識是提高軟體的品質的一項重要措施。其內容包括保證專案滿足原先規定的各項要求，所需實施的組織活動。即決定品質方針，目標與責任的所有活動。並通過諸如品質規劃，品質

保證，品質持續改進等方針，程序和過程來實施品質體系。較著名的相關軟體開發管理方法與知識有 CMMI、PMBOK、IEEE 開發流程等。

(一)能力成熟度整合模式

在 2000 年底時美國卡內基美隆軟體工程學院(Carnegie Mellon Software Engineering Institute, SEI)發表了能力成熟度模式整合 CMMI。CMMI 是軟體工程具體化的最佳例子，也是一個軟體開發標準流程，目的在發展一個共通性之整合架構，以支援整合不同專業領域之特定能力成熟度模式及相關產品，並致力提供系統工程及軟體工程之指導原則，期許在任何架構下的組織，皆能促進其流程改善，CMMI 不僅提高每一級別成熟度要求之門檻，亦同時擴充能力成熟度評鑑適用範圍，使得軟體工程、系統工程之專業領域及整合性產品與流程發展之環境，皆能運用 CMMI 為軟體開發流程提供持續改善的指引，對軟體生產力與品質的提升亦有顯著的實質效益，並確保所有發展的產品，能與國際標準組織／國際電技協會(ISO／IEC) 15504 軟體流程評鑑技術報告相容並一致。在每一個流程領域中 CMMI 定義了特定目標(Specific Goal, SG)及一般目標(Generic Goal, GG)，此特徵用來說明必須要執行甚麼活動以滿足該流程領域。以 CMMI 在能力成熟度第二級中的流程與產品品質保證流程為例，品質保證工作應開始於專案建立的初期，參與建立專案的計畫、流程、標準及程序，為專案增加價值，並滿足專案和組織政策的需求，該流程領域定義了特定目標(Specific Goal, SG)及一般目標(Generic Goal, GG)，分別是：SG 1 是客觀評估流程與工作產品、SG 2 是提供客觀的洞察力、GG 2 是制度化已管理流程與 GG 3 是制度化已定義流程。

(二)專案管理知識體系

PMBOK 是美國項目管理學會的 PMI 對於專案管理職業的知識總和，並將專案過程的每一個活動以系統的方式，定義出每一個活動的輸入(Inputs)和產出(Outputs)，由實際的工作者和學者來逐步

建立 (PMI Standards Committee, 2004)，共包含了四十四個專案管理過程，並可分類為九大管理知識領域及五種專案管理流程組。其中九大知識領域分別是專案整體管理 (Integration Management)、專案範圍管理 (Scope Management)、專案時間管理 (Time Management)、專案費用管理 (Cost Management)、專案品質管理 (Quality Management)、專案人力資源管理 (Human Resource Management)、專案溝通管理 (Communications Management)、專案風險管理 (Risk Management)與專案採購管理 (Procurement Management)。而 PMBOK 將專案管理的各個過程歸納為五大程序則為啟動、規劃、執行、監控、結案。

(三) IEEE 軟體專案標準作業流程

IEEE 的標準化活動主要涉入軟體專案、資訊技術等相關領域。根據 IEEE Standard Board (1987)將軟體專案定義如下：「有關一系列的專案職務、任務、活動等技術與管理工作，以滿足專案協議中所需的術語與情況；軟體專案本身可以是一個完整的計劃或者是大型計劃的一部份，並涉及一部份的軟體產品生命週期。」一般而言，一個軟體開發流程是在描述軟體開發活動中，需進行哪些工作、如何進行、由誰執行、何時執行，並提供一些指導準則供軟體開發人員參考 (陳利銘, 2003)。軟體開發流程即是用以指導、管理軟體開發過程中所涉及之人、專案、產品與流程 (Kruchten, 2000)。負責軟體專案的個人或組織亦應負責軟體專案管理計畫書(簡稱為 SPMP)。本標準的本項條款在說明 SPMP 的每項要素。另外學者 Kruchten 也說明軟體開發流程在整個軟體開發中扮演著以下重要角色：(1)指導軟體專案團隊之軟體開發活動，(2)指導在軟體開發過程中所應產生的產出物 (Artifact)，(3)指導軟體開發人員與團隊的工作成為一體，與(4)針對監控與衡量專案產品與活動，提供一個指導架構。

以下為本研究依據軟體專案開發標準整理比較：

表 1 專案軟體開發標準的比較

比較項目	軟體專案開標準		
	CMMI	PMBOK	IEEE
來源	美國卡內基美隆軟體工程學院	美國項目管理學會 PMI	美國電機電子工程師學會
重點	以共通性之整合架構提供系統工程及軟體工程之指導原則，促進其流程改善。	定義明確的專案管理知識體系和方法論。提供基本知識，執行單一專案時的依循標準。	定義專案職務、任務、活動等技術與管理工作，以滿足專案協議中所需的術語與情況。
分類	定義各流程領域特定目標 (Specific Goal, SG) 及一般目標 (Generic Goal, GG)。	九大管理知識領域、五種專案管理流程、五大專案管理程序。	IEEE 軟體專案管理計畫 (SPMP) 標準。
應用	軟體開發流程的評估。	專案開發過程知識應用。	專業軟體工程作業標準。

(資料來源：本研究整理)

2.2 軟體專案的知識管理

基於軟體專案開發涉及複雜文件與資源的管理，需要一套知識分類的理論作為知識的分類依據，為了達成此一目的，學者 Swartout 與 Tate (1999)認為本體論可作為知識表達的基礎，避免重複的領域知識分析，藉由統一的概念及術語來達成知識之共享，本體論還可以幫助知識正規化並與組織人員分享 (Staab et al., 2001)。學者劉文卿與馮國卿 (2003)認為建構知識本體論的功用是分享該領域內有哪些重要的觀念，所定義的名詞間是以什麼關係連結，藉由分享的機制，每一個人都知道這個領域內字彙組織的脈絡。利用這些字彙來發展我們的應用系統，其優點是：字彙的一致性能夠提昇語意關聯性搜尋的準確性，且基於相同的溝通機制能與其它運用相同領域字彙所發展的系統溝通，達到有效的名詞分享。對於知識工程而言，本體論主要協助知識庫系統之建立，以塑模的方法

來呈現所有的事實或概念，並以特定的塑模詞句來達到正確的概念表示 (Greenhill and Venkatesh, 2000)，並幫助專案開發人員進溝通、搜尋、存取及描述表示之目的 (O'Leary, 1998)。

以軟體專案工作的內容而言，其屬於高度知識密集的工作，不論專案管理者或專案開發人員，皆需要完整、豐富的知識以協助專案順利進行。因此，一套適合於軟體專案的知識本體論即可創造下列的優勢如下：(1)增加組織整體知識的存量與價值，(2)應用知識以提升技術、產品、服務創新的績效以及組織整體對外的競爭力，(3)促進組織內部的知識流通，提升成員獲取知識的效率，(4)指導組織知識創新的方向，(5)協助組織發展核心技術能力，(6)有效發揮組織內個體成員的知識能力與開發潛能，(7)提升個體與組織整體的知識學習能力，與 (8)形成有利於知識分享的企業文化與價值觀。

2.3 知識本體論的應用

知識本體論最早是在哲學的領域中被提出來，主要探討什麼是已存在的實體 (entity) 或事物 (Things)，以及這些實體間是以什麼關係來交互作用，並嘗試說明如何定義並規範其表示的方式，而且隨著目前電腦科技的發展，知識本體論已漸漸從人工智慧上應用到知識管理之領域，用以分享知識和再使用知識 (劉文卿、馮國卿，2003)，基本上，本體論主要區分為以下各類：(1)頂級本體論 (Top Level Ontologies): 描述最普遍的概念及概念之間的關係，如時間、空間、事件、行等等，與具體的應用無關，(2)領域本體論 (Domain Ontologies): 指知識的取得是經由特別領域的確認，(3)任務本體論 (Task Ontologies): 指從特定任務或行為所發展出的作業方法知識，與 (4)應用本體論 (Application Ontologies): 指要塑模特別領域所必須有的知識，通常是領域本體論與任務本體論的結合 (Guarino, 1998)。

根據 Gruber (1993) 將本體論定義為「本體論是一種概念模型的詳細規範說明。」對於知識工程而言，運用本體論之主要目

地為協助知識庫系統之建立，以塑模的方法來呈現所有的事實或概念，並以特定的塑模詞句來達到正確的概念表示 (Greenhill and Venkatesh, 2000)，以幫助專案開發人員進溝通、搜尋、存取、描述與表示知識 (O'Leary, 1998)。

以軟體專案工作的內容而言，其屬於高度知識密集的工作，不論專案管理者或專案開發人員，皆需要完整、豐富的知識以協助專案順利進行。因此，一套適合於軟體專案的知識本體論即可創造下列的優勢：(1)增加組織整體知識的存量與價值，(2)應用知識以提升技術、產品、服務創新的績效以及組織整體對外的競爭力，(3)促進組織內部的知識流通，提升成員獲取知識的效率，(4)指導組織知識創新的方向，(5)協助組織發展核心技術能力，(6)有效發揮組織內個體成員的知識能力與開發潛能，(7)提升個體與組織整體的知識學習能力，與 (8)形成有利於知識分享的企業文化與價值觀 (Gruber, 1993；余明真，2005)。

2.4 物件導向方法應用在知識庫之應用

知識庫之建置多半採用物件導向 (Object-Oriented Thinking) 技術設計，配合統一塑模語言 (United Modeling Language, UML) 之圖形工具以作為設計規劃的藍圖及依據。而本體論原是一個哲學的分支，探討客觀事物存在的本質。它與認識論 (Epistemology) 相對，認識論研究人類知識的本質和來源。也就是說，本體論研究客觀存在，認識論研究主觀認知。而本體論的含義是形成現象的根本實體。

在知識庫系統中，知識建模必須在知識庫和兩個子系統，語意庫與規則庫之間建立聯繫。大規模的模型共用、系統組合、知識擷取和再利用 (Reuse) 都有賴於領域的知識結構分析。知識庫乃是存知識的電腦資料庫，與資料庫之不同處，在於知識庫是從資料庫的模式衍生而來。即是將專家腦中非結構化或難以文字、數字、圖案、表格、公式... 等表達的專業知識與經驗，以法則及事實 (Fact) 的型式儲存於資料庫之中。因為知識庫裡的資訊會成為將來知識應用推論引擎比對的依據，同時也是知

識擷取引擎擷取知識後，所存放的地方。因此，知識庫內的規則與事實必須是清楚且正確。在建立知識庫時，其中的欄位都是和專家討論過後所得到，因此這些欄位能作為形容知識庫中的法則與事實的特徵 (Feature)，而這些特徵的集合便能代表知識庫中的某一個案例。

一般來說，本體論提供一組術語和概念來描述某個領域，知識庫則使用這些術語來表達該領域的事實。實際上，如果本體論和知識庫用同一語言表達的語，兩者之間並沒有清晰的界線。區別僅僅在於知識庫的哪一部分是可以分享和再利用的，哪一部分是針對特定應用的，這種區別往往還隨著時空因素的具體背景變化。從描述物件的類型來說，本體論既可以用來描述簡單的事實，又可以用來描述信念、假設、預測等抽象的概念；既可以描述靜態的實體，又可以描述與時間推移相關的概念，如事件、活動、過程等。從描述物件的範圍來說，本體論可以定義通用的、適合所有領域知識表示的術語，如空間、時間、部分等；也可以定義特定領域知識才使用的術語。不同本體之間存在著差別，但它們在較高的抽象層次上 (Upper Ontology) 具有一些共同的特徵。從客觀的意義上說，本體的描述是和特定的任務和目的無關的。但是我們使用本體總是有一定的任務背景，因此對於所描述的知識的選擇是和特定任務相關的。本體論作為通訊、互操作和系統工程的基礎，必須經過精心的設計，基本上本體構造依據五個準則，即：(1) 清晰性 (Clarity)，(2) 一致性 (Coherence)，(3) 可擴展性 (Coherence)，(4) 編碼偏好程度最小化 (Minimal Encoding Bias)，與(5) 本體約定最小化 (Minimal Ontological Commitment)。Uschold & Gruninger (1993) 也提供了一個本體構造的方法學論架構，該架構包括以下組成部分：(1) 確定本體的目的和使用範圍，(2) 構造本體，(3) 文檔記錄，與(4) 每一階段的指導準則。

總而言之，知識庫本體在一些商用系統中得到應用，它的重要性已經被異構資料

庫集成、知識工程和規劃等不同領域所認識。目前的主要困難在於：儘管構造本體的目的在於再利用，但真正再利用的實例卻不多見。人們往往喜歡構造自己的本體，造成這種情況的原因主要有兩方面，其一是本體構造目的不同；其二是本體表示方法不統一，而本研究實作知識庫的建置就是要解決這個問題，藉著統合的教學目標和教師共通知識，達到知識的再利用。

3. 以本體論為基礎的軟體專案知識庫系統架構

本研究主要是探討軟體專案團隊在專案開發各流程階段的中透過以本體論觀點建構的軟體專案知識庫系統進行知識管理與知識分享，期能提升各團隊在軟體開發流程中的效率與知識分享，以期能夠準確的提升各團隊在軟體開發中的績效。有鑑於此，本研究綜合前述文獻之分析進行本研究架構之建立。本研究主要是由下列各部分所構成：(1) 軟體專案開發流程，(2) 軟體專案人員與單位使用人員，(3) 軟體專案知識庫系統，與(4) 知識庫應用績效，如圖 1 所示。「軟體專案開發流程」即表示一個軟體專案從執行到結案的整個開發活動之程序；「軟體專案人員與單位使用人員」主要是所有使用軟體專案知識庫系統的人員；「以本體論為基礎的軟體專案知識庫系統」即代表分類及貯存整個軟體專案的領域知識，協助軟體專案人員有共同的術語有效進行軟體專案開發之活動；「知識庫應用績效」軟體專案人員在使用軟體專案知識庫系統後分享使用績效與討論改善，構成知識分享的行為，以滿足專案團隊人員對於知識的需求，進而提升知識庫應用在軟體專案開發之績效。

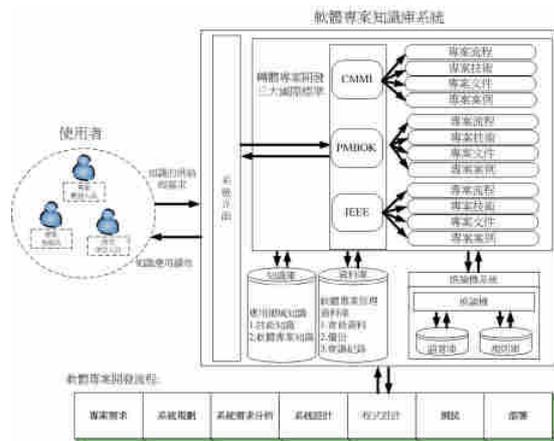


圖 1 以本體論為基礎的軟體專案知識庫系統架構
(資料來源：本研究整理)

(1) 軟體專案開發流程

由於一般軟體專案之開發流程，從起始專案規劃階段至驗收與正式維護階段為止，均有一定標準程序與進行步驟。因此，本研究將探討至整個軟體開發流程中各階段之知識分享活動，依據相關文獻 (Boehm, 1976; Moller and Paulish, 1993; Peter, 1998; 陳利銘, 2003; 湯美蓮, 2004; 黃明祥, 2005) 歸納整理，主要是軟體專案開發生命週期，軟體開發流程依序為：專案規劃階段、專案設計階段、軟體需求分析階段、軟體設計階段、軟體製作階段、軟體測試階段與驗收與正式作業階段。

(2) 軟體專案人員與單位使用人員

使用者於初次使用軟體專案知識庫系統時將會透過知識分享的方式來獲取包含軟體專案的開發流程、相關文件與技能、使用工具等關於軟體開發的專業知識描述資訊，並將其儲存在軟體專案知識庫中，提供後續使用者開發軟體專案規劃時的依據，屬於本研究之重要目的。本研究使用者主要人員有三種：專案管理員、專案開發人員，與專案使用人員。

(3) 以本體論為基礎的軟體專案知識庫系統

軟體專案的知識所涵蓋知識的種類相當廣泛。因此，本研究運用本體論主要的原因為：本體論在知識管理領域中被運用的主要原因之一為達成某一領域知識有共通的表示方法、術語並清楚的表達概念，其主要的目的是達到知識的分享與再使用等目標。基本上，本體論將某一領域知識

分解成數個知識元件，並加以定義與描述元件與元件之間的關係，而描述的方法及呈現的方式都需視應用的領域而決定。本研究所制定的本體論主要是在協助解決軟體專案知識的分類問題，而發展一個適用於軟體專案的知識分類方法，以提升知識庫系統的知識分享的效益，讓知識庫系統能夠有效的滿足軟體專案團隊的知識需求。

(a) 以本體論為基礎的知識分類

有鑑於此，本研究所定義的本體論主要是針對軟體專案開發的知識領域，依據軟體專案的知識管理之理論，提出一個以本體論為基礎的軟體專案知識分類架構，如圖 2 所示，其中透過將各個概念拆成數個子節點 (概念)，運用樹狀的方式來呈現各個概念以及概念與概念之間的關係，而每個概念又代表著軟體專案開發所運用到的領域知識。首先是由軟體專案知識領域此節點開始往下細分，包括專案開發知識三大標準應用領域知識 (專案管理知識體系, 2004)，此節點又往下細分：知識本體、專案開發文件知識 (陳利銘, 2003; 湯美蓮, 2004; 黃明祥, 2005)，專案技術知識 (軟體工程知識體系, 2004) 與專案案例等。

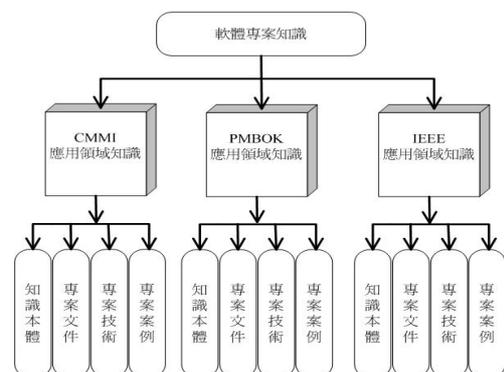


圖 2 以本體論為基礎的軟體專案知識分類方法
(資料來源：本研究整理)

(b) 知識庫

知識庫 (Knowledge Base) 是本研究之系統核心，而豐富之知識事實與合理的推論規則，才能造就一個有價值的知識庫系統，配合使用者需求，知識庫的主要應用領域知識內容包括軟體專案技術知識、軟體專案知識等，以提供系統能在各種設定條件下，選擇應有之功能。知識庫將協助

系統使其能直接與使用者互動，使所有的資料、圖表、各種參數等資料能夠適當的傳送、整理、更新等。本研究設定使用者會依據權限而分別擁有不同之介面功能，操作上則以 Web 介面的瀏覽器 (Browser) 為主，透過瀏覽器以 HTTP 的方式遠端登入至系統中進行操作知識管理活動，一般常見的知識管理活動有知識的建檔、分類、貯存、擷取、查詢、過濾與推薦等。知識庫主要用於貯存目前所有的軟體專案相關知識，專案執行期間所產生的新知識亦同樣亦貯存於知識庫中。知識庫中的知識是依照以本體論為基礎的軟體專案知識分類方法所進行分類與貯存。

(c) 推論機系統

推論機系統主要是強化知識搜尋，輔助知識庫系統，推論機可仿人類解決問題的技巧及方法，以知識庫裡所儲存的專門知識來解答問題，即使不具豐富經驗之一般操作員也能據此下判斷，而且能顯示推論的過程。知識庫系統透過電腦輔助，擷取人類專家知識，在資訊提供較不完整情況下，仍可透過推論機之內建推理邏輯，具備合理推算能力，同時在面對多變的交通環境及複雜的幾何條件下，亦能隨時藉由知識庫的擴充，以傳遞、轉換及推論的過程，減少數學求解負擔，即時提供近似解。本研究之推論機系統包含推論機、語意庫、規則庫等三個主要部份：(1) 推論機 (Inference Engine)：負責處理規則的推論運算，透過向前推論或向後推論的方式，以推導出新的事實 (Fact)，可執行高度複雜及動態推論過程，在動態控制過程中。其中工作記憶區 (Working Memory) 是儲存所有工作階段中的運算資料，其資料為暫時性而非永久性，當系統關機或重新載入時，資料便會消去，(2) 語意庫 (Semantic Base)：把軟體專案資料的意義定義的更明確，使知識庫系統更有效率。由於軟體專案資訊太過龐大與繁雜，所有的資料都是以文件的方式所呈現。語意庫將文件有意義的結構化，以建立起一個資訊充份分享及知識可重覆使用的空間。藉由語意庫，當人們要查詢資料的實際函義時，可以透

過超連結找到關鍵字的定義，並使用法則 (Rules) 來邏輯的推論 (Berners-Lee & Fischetti, 1999)，與(3)規則庫 (Rule Base)：規則庫儲存所有以規則呈現的內容，規則庫中的規則是由條件 (Condition) 和動作 (Action) 所組成，條件規則的形式則是以 If-Then-Else 的方式表達。動作部分必須判斷何時該執行，何時不該執行，而這些判斷的條件全都存放在規則庫中，當比對到某條規則中的條件 (Condition) 符合時，就會觸發執行相對應到的工作 (Action)。本研究之知識庫系統會將分析的結果作規則比對，經過語意庫與規則庫中的規則比對之後，回應使用者相對應的引導，以達到輔助的功能。

(d) 知識庫系統之應用績效

軟體專案是一項由多人，包括開發人員與使用者共同合作完成的智慧資產，在各階段的軟體專案開發過程中，專案人員通常需要多樣化的專案知識以協助軟體專案開發與管理工作之進行，使用者則要於使用後分享使用績效與討論改善。因此，本研究架構其應用範疇涵蓋整個軟體專案的開發過程，在每一個專案開發階段中，專案團隊人員對於軟體專案的知識有供給與需求，構成知識分享的行為，並且透過以本體論為基礎的軟體專案知識庫系統實做出一套可供知識分享活動的平台，包含知識的建檔、分類、貯存、擷取、查詢，以滿足專案團隊人員對於知識的需求，進而提升知識庫應用在軟體專案開發之績效

。本研究擬用軟體成本度量作為績效評估對象，其主要在於蒐集軟體開發過程中完成一個工作項目或一個階段所耗費的實際工作量，並且能夠真實反應實際的成本度量值；但除了度量值的蒐集外，尚須有預先估計之成本估計值做為績效衡量指標的基準。因此，軟體成本的估計是軟體成本度量工作中首要進行的步驟，而成本預估工作主要取決在人力 (Manpower)、工作量 (Effort) 和時程 (Schedule) 等三個項目 (Vigder and Kark, 1994)。由於軟體發展方法與工具以及電腦輔助軟體工程工具的不斷開發與進步，使得軟體成本估計工作愈

形複雜。有鑑於此，在軟體專案管理知識當中，成本估計模式應用十分廣泛，本研究擬以「成本估計模式」知識收尋做為範例(如圖 8, 9)。

Boehm 等人於 1995 年針對原有的 COCOMO 成本估計模型再度進行修正，產出 COCOMO II 的新版本 (Boehm et al., 1995)。COCOMO II 改善了原本 COCOMO 只適用於瀑布式開發流程的缺點，而針對現代軟體開發流程，提出反覆式和漸進式估計的概念，使估計的結果能隨著專案的進行而提高正確性。同時，在實務應用方面，由於 COCOMO II 具有適用性廣、容易使用和過程透明等優點，因此，COCOMO II 適合應用於中大型軟體專案之成本與時程估計工作 (Boehm et al., 1995)。

在軟體專案進行過程中，除了須持續蒐集實際成本度量值外，為能夠對專案成本作進一步的績效控管，差異分析法 (Variance Analysis) 則是歷史最悠久也是應用最普遍的一種衡量專案績效的方法 (Saad, 2003)。差異分析法具有簡單又容易使用的特性，透過專案預估值與實際值之差異比較，即可獲得專案成本與時程之績效 (Saad, 2003)。再者，根據黃明祥與盧煥文 (2006) 之研究發現，差異分析法應用於實務之專案控管效果良好。由此可知，COCOMO II 估計模式與差異分析法經常被應用在軟體專案成本與時程之估計及控管工作。

4. 軟體專案知識庫系統設計與分析

4.1 個案背景分析

本研究係以國內某國立科技大學的軟體專案管理研究中心作為實證對象，該個案有鑑於下列問題：(1) 軟體開發工作屬於人力、知識密集之工作，一旦開發人力流失便代表著知識的流失，因而意識到知識管理的重要性，(2) 跨領域與委託專案增加，團隊合作與協同工作之需求增加，(3) 保留軟體開發過程中之重要文件與資料，以作為往後軟體專案開發之知識管理基

礎，與(4)提升軟體品質與軟體開發人員之生產力。為了解決上述問題，本研究乃依據以本體論為基礎建立軟體專案知識庫系統，以協助專案團隊進行知識管理活動並且有效支援各階段軟體開發工作之進行。

4.2 系統架構

社會中人們以不同的方式取得資料與資訊，並使其成為有用的知識。在智慧型系統的研究領域中，愈來愈多的研究在發展知識庫幫助使用者從資訊中取得所想要的知識，尤其在高度知識密集與需求的軟體專案更是迫切需要一套知識庫系統管理軟體專案團對知識的傳遞、分享與再利用，組織收集及整理資訊，並把資訊快速的傳送給需要的人，持續分析以維護資訊有效性的過程與機制。為了解決上述問題，本研究擬發展一套以本體論為基礎的軟體專案知識庫系統，協助專案團隊人員於軟體專案開發環境下提供專案人員在軟體開發過程中所需的知識，進而提昇專案開發的績效。本研究所發展的軟體專案知識庫系統其完整架構如圖 3 所示，該系統主要涵蓋以下部分：(一) 軟體專案知識庫系統：包括軟體專案知識庫系統介面、知識分享活動、知識庫與軟體專案資料庫，(二) 推論系統：包含推論規則擷取介面、工作記憶區、推論機、規則庫、語意庫等。

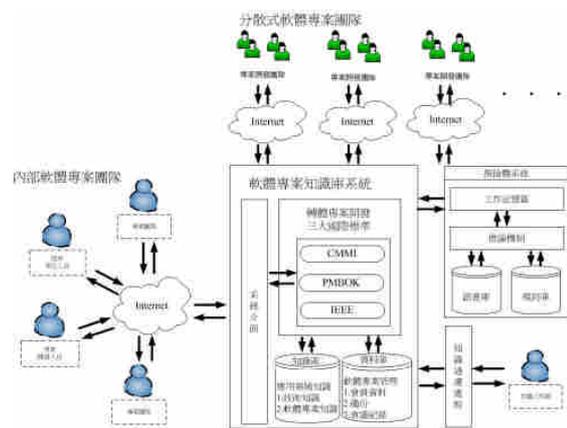


圖 3 系統架構圖

(資料來源：本研究整理)

以下對系統架構中各個元件之功能進行說明：

(1) 軟體專案知識庫系統介面：本系統

所針對的使用者對象是專案管理者、專案開發人員、知識工程師與專案客戶等，其分別擁有不同的系統權限，操作上則以 Web 介面的瀏覽器 (Browser) 為主，使用者可以透過瀏覽器以 HTTP 的方式遠端登入至系統中進行操作。(2)使用者開發團隊：本系統主用者有軟體專案管理者、軟體專案開發人員、軟體專案人員與知識工程師等角色。當各個使用者欲透過本系統進行軟體專案開發時，可透過軟體專案知識庫系統介面進行操作。(3)推論系統：推論系統主要分為以下幾個功能(a)推論機：主要負責處理知識規則的推論運算，透過向前推論或向後推論的方式，以推導出新的事實 (Fact)。(b)語意庫：主要是把軟體專案資料的意義定義更明確，將文件結構化，以建立起資訊充份分享及知識可重覆使用的空間，使知識庫系統更有效率。(c)規則庫：規則庫中儲存所有以規則呈現的內容，規則庫中的規則是由條件 (Condition)和動作 (Action)所組成，條件規則的形式則是以 If-Then-Else 的方式表達。(4)知識庫：用來貯存專案所有的知識與知識規則，專案執行期間產生的新知識也會儲存於知識庫中。

4.3 範例展示

本研究實證分析對象為國內某國立科技大學之軟體專案管理研究中心為例，經過需求分析，將軟體專案知識庫系統的使用者依職等與權限的不同分為專案管理者、專案開發人員、知識工程師與專案使用者，透過系統使用權限之控管，制定出不同使用者的權限。為了瞭解運用本系統實際支援軟體專案運作的情形，本研究係以專案人員當進行軟體測試階段時之例子進行說明。首先，專案人員需先進行登入動作(圖 4)，檢核其帳號權限，若通過身份認證後，即可進入該系統。圖 5 為專案人員選擇軟體專案知識分類規則或軟體專案知識查詢之功能擷取軟體測試所需知識(圖 6)。在進行軟體測試過程中若有新知識的產生，可運用知識建檔之功能(圖 7)，將軟體專案知識有效累積貯存。

若專案人員在軟體測試的階段中需要

參考先前專案所完成的軟體測試說明書以及本專案的軟體需求分析規格書，即可透過軟體專案文件管理功能，依據專案代號或文件名稱進行查詢(圖 8)，例如用 COCOMO 作為查詢之範例，其查詢結果將會如圖 9 所示，可供點閱或下載，其效率比無本體論分類架構(圖 10)之搜尋精準與快速。



圖 4 系統使用者身份認證

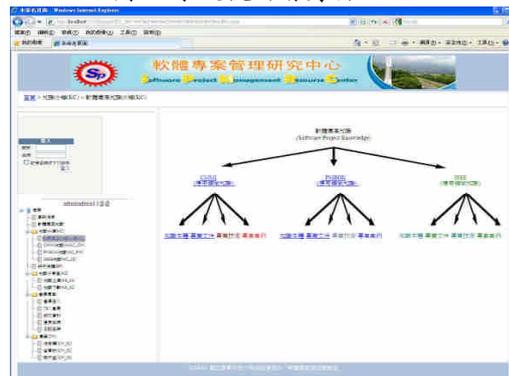


圖 5 知識查詢—依本體論之分類規則



圖 6 軟體測試知識的查詢結果



圖 7 軟體專案知識建檔



圖 8 軟體專案文件查詢—依文件名稱

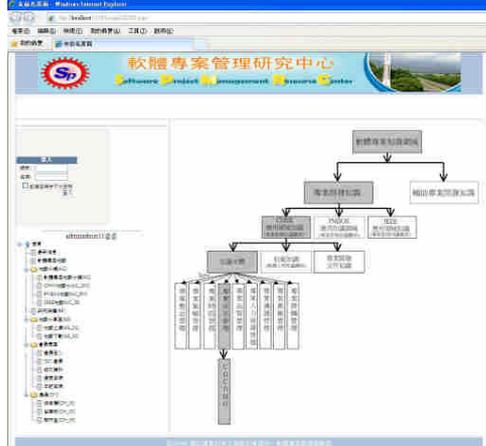


圖 9 有本體論分類架構之軟體專案文件查詢



圖 10 無本體論分類架構之軟體專案文件查詢

5. 結論

本研究所建立的以本體論為基礎的軟

體專案知識庫系統之實作，具有以下特色與優點：(1)本系統是以 Web 為基礎的系統操作介面，可提供分散式開發環境之下的各個專案團隊人員使用，(2)根據軟體專案管理之特性，發展一個以本體論為基礎的知識分類架構作為本系統知識分類的標準依據，(3)依據各別的使用者給予不同的系統權限，有效落實知識分享之安全性控管，與(4)運用本系統以協助專案團隊進行知識管理活動，並且有效支援各階段軟體開發工作之進行。藉由以下各項觀點說明應用本系統的效益：(1)以軟體專案管理觀點而言：能夠支援各階段的軟體專案開發流程，有效提升軟體專案開發品質與生產力，(2)以知識管理觀點而言：達成知識管理的活動，將軟體知識進行有效的擷取、組織、精煉與分佈，與(3)以技術觀點而言：透過本體論的理論實作之系統能夠在分散式的軟體開發環境中，有效進行知識的擷取、分享與再使用，並有效控管知識分享之安全性。

綜合以上所述，本研究主要的貢獻可大致分為學術與實務兩方面，學術方面的貢獻為探討專案管理知識庫應用在軟體專案開發之重要議題、相關理論與應用技術建構出本研究架構，進而實作出一個軟體專案管理的知識庫系統，與依據本體論發展出一個適用於軟體專案領域知識的分類方法，建構出以本體論為基礎的專案管理知識庫，供相關人員進行軟體專案團隊知識分享相關研究之參考用途。在實務方面的貢獻，本研究發展一套以本體論為基礎的軟體專案知識庫系統，提供軟體專案團隊在知識分享的輔助工具並支援軟體開發流程之進行，以及不斷地累積軟體專案開發過程之經驗與技術，提升知識的分享與再用性，作為實務界建立與應用相關知識庫系統之參考依據。

參考文獻

- [1]梁定澎、歐陽彥晶與許如欽等著，2005，影響台灣企業採用知識管理之因素，資訊管理學報，第十二卷，第三期。

- [2]戴明，1999，戴明的新經濟觀，天下遠見出版社。
- [3]湯美蓮，2004，以 Web 為基礎之軟體專案進度監控資訊系統之建立與應用，國立屏東科技大學資訊管理系碩士論文。
- [4]林信惠、黃明祥與王文良等著，2002，軟體專案管理，智勝文化事業有限公司。
- [5]林東清，2004，知識管理，智勝文化事業有限公司。
- [6]林東清，2005，資訊管理—e 化企業的核心競爭能力，智勝文化事業有限公司。
- [7]劉文卿、馮國卿，2003，以標準化 Metadata 為核心發展金融機構 Ontology 之探討，國書館學與資訊科學，第 45-59 頁。
- [8]黃明祥，2005，物件導向軟體專案開發制度之建立與導入—以國內鋼鐵公司之研發部門為例，資訊管理學報，第十二卷，第一期，第 57-89 頁。
- [9]黃明祥、盧煥文，2006，「以智慧型代理人為基礎的知識庫系統在軟體專案成本估計與控制之應用」，資訊管理學報，第十三卷·第一期：137~167 頁。
- [10]黃文宏，2001，知識管理理論與 UML-based 軟體開發實務結合之初步探討—以鋼鐵製造業個案為例，國立屏東科技大學資訊管理系碩士論文。
- [11]莊淇銘，2006，「學習」已經落伍了！：掌握知識管理才是贏家，新自然主義。
- [12]張紅延、黃濤、蔣明青，2004，軟件質量度量及其模形指標的研究，計算機應用研究 21 卷 10 期。
- [13]周斯畏，2001，物件導向系統分析與設計：使用 UML 與 C++，全華科技圖書。
- [14]陳利銘，2003，以 Web 為基礎之軟體專案開發流程模式之建立—以國內大型鋼鐵公司研發部門為例，國立屏東科技大學資訊管理系碩士論文。
- [15]楊政學，2005，知識管理學理與實證，揚智文化。
- [16]俞宣孟，2005，本體論研究，上海人民。
- [17]謝清佳，2003，知識管理，智勝文化事業有限公司。
- [18]Abran, A., Moore, J., Bourque, P., Dupuis, R. and Tripp, L., "Guide to the Software Engineering Body of Knowledge," IEEE-Computer Society, 2004.
- [19]Alavi, M., "Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," MIS Quarterly, Vol. 25, No. 1, 2001, pp.107-136.
- [20]Albert, L.K., "YMIR: An Ontology for Engineering Design," Ph.D. Thesis, University of Twente, Twente, The Netherlands, 1993.
- [21]Aurum, A., Jeffery, R., Wohlin, C. and Handzic, M., "Managing Software Engineering Knowledge," Springer, Germany, 2003.
- [22]Awad, E.M. and Ghaziri, H.M., "Knowledge Management," Prentice-Hall, 2004.
- [23]Barnes, S., "Knowledge Management Systems: Theory and Practice," Thomson Learning, 2002.
- [24]Berners-Lee, T. and Fischetti, M., Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor, Harper Business, 1st edition, San Francisco, 1999.
- [25]Boehm, B.W., "Software Engineering," IEEE Transactions on Computer, Vol. C-25, No. 12, 1976, pp.1226-1241.
- [26]Boehm, B., Clar, B., Horowitz, E., Westland, C., Madachy, R. and Selby, R. "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," Ann. Software Eng., Volume: 1, No: 1, 1995, pp: 1-30.
- [27]Boehm, B., Chulani, S. and Clark, B. "Calibrating the COCOMO II Post Architecture Model," <http://sunset.usc.edu/publications/TECHRPTS/1998/usccse98-502/CalPostArch.pdf>, 1997.
- [28]Boehm, B., Software Cost Estimation

- with COCOMO II, Prentice Hall PTR, 2000.
- [29] Bollinger, A.S. and Smith, R.D., "Managing Organizational Knowledge as A Strategic Asset," *Journal of Knowledge Management*, Vol. 5, No. 1, 2001, pp.8-18.
- [30] Davenport, T.H. and Prusak, L., "Working Knowledge: How Organizations manage What They Know," Harvard Business School Press: Boston MA, 2000.
- [31] Davis, J. G., "The Global and The Local in Knowledge Management," *Journal of Knowledge Management*, Vol. 9, No. 1, 2005, pp.101-112.
- [32] Desouza, K.C. and Evaristo, J.R., "Managing Knowledge in Distributed Projects," *Communications of the ACM*, Vol. 47, Iss. 4, 2004, pp.87-91.
- [33] FIPA ACL Message Representation in XML Specification, <http://www.fipa.org/specs/fipa00071>, 2001.
- [34] Greenhill, S. and Venkatesh, S., "Semantic Data Modeling and Visualization Using Noetica," *Data & Knowledge Engineering*, Vol. 33, Iss. 3, 2000, pp.241-276.
- [35] Guarino, N. and Giaretta, P., "Ontologies and Knowledge Bases: Towards A Terminological Clarification," IOS Press, 1995, pp.25-32.
- [36] Guarino, N., "Formal Ontology and Information Systems," *Proceedings of FOIS'98, Trento, Italy, Amsterdam*, 1998, pp.3-15.
- [37] IEEE STANDARDS BOARD, IEEE Standard for Software Project management Plans IEEE Std. 1058.1, 1987.
- [38] Kettunen, P., "Managing Embedded Software Project Team Knowledge," *IEE Proceedings on Software*, Vol. 150, Iss. 6, 2003, pp.359-366.
- [39] Knight, T. and Howes, T., "Knowledge Management- A Blueprint for Delivery," Butterworth Heinemann, 2002.
- [40] Kruchten, P., "The Rational Unified Process: An Introduction," Second Edition, New York: Addison-Wesley, 2000.
- [41] Liebowitz, J., "Knowledge Management and Its Link to Artificial Intelligence," *Expert Systems with Applications*, Vol. 20, Iss. 1, 2001, pp.1-6.
- [42] Moller, K.H. and Paulish, D.J., "Software Metrics- A Practitioner's Guide to Improved Product Development," Chapman & Hall Pub. 1993, pp.257.
- [43] O'Leary, D.E., "Using AI in Knowledge Management: Knowledge Bases and Ontologies," *IEEE Intelligent Systems*, Vol. 13, Iss. 3, 1998, pp.34-39.
- [44] O'Leary, D.E., "Enterprise Knowledge Management," *IEEE Computer*, Vol. 31, Iss. 3, 1998, pp.56-61.
- [45] Peter, L., "Advanced Structured Analysis and Design," Prentice-Hall, 1998.
- [46] PMI (Project Management Institute) Standards Committee, "A Guide to The Project Management Body of Knowledge," Pennsylvania:Project Management Institute, 2004.
- [47] Probst, G., Raub, S. and Romhardt, K., "Managing Knowledge- Building Blocks for Success," John Wiley & Sons Ltd: Chichester, 2000.
- [48] Rakos, J.J., "Software Project Management for Small to Medium Sized Projects," Prentice Hall, 1990.
- [49] Ruppel, C.P. and Harrington, S.J., "KM on Intranets and The Fifth Dimension," *IEEE Potentials*, Vol. 20, Iss. 2, 2001, pp.19-24.
- [50] Saad, H. A., "Monitoring Systems and Their Effectiveness for Project Cost Control in Construction," *International Journal of Project Management*, 2003, pp: 145-154.
- [51] Satyadas, A., Harigopal, U. and Cassaigne, N.P., "Knowledge Management Tutorial: An Editorial Overview," *IEEE Transactions on Systems, Man., and Cybernetics*, Vol. 31, Iss. 4, 2001, pp.429-437.
- [52] Staab, S., Studer, R., Schnurr, H.P., and

- Sure, Y., "Knowledge Processes and Ontologies," IEEE Intelligent Systems, Vol. 16, Iss. 1, 2001, pp.26-34.
- [53] Storck, J., "Knowledge Diffusion Through Strategic Communities," Sloan Management Review, Vol. 41, No. 2, 2000, pp.63-74.
- [54] Studer, R., Benjamins, V.R. and Fensel, D., "Knowledge Engineering: Principles and Methods," IEEE Transactions on Data and Knowledge Engineering, 1998, pp.161-197.
- [55] Sunassee, N.N. and Sewry, D. A., "An Investigation of Knowledge Management Implementation Strategies," Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology SAICSIT '03, 2003, pp.24-36.
- [56] Swartout, B., Patil, R., Knight, K. and Russ, T., "Toward Distributed Used of Large-scale Ontologies," Ontological engineering, 1997, pp.138-148.
- [57] Swartout, W. and Tate, A., "Ontologies," IEEE Intelligent Systems, Vol. 14, Iss. 1, 1999, pp.18-19.
- [58] Tiwana, A., "An Empirical Study of The Effect of Knowledge Integration on Software Development Performance," Information and Software Technology, Vol. 46, Iss. 13, 2004, pp.899-906.
- [59] Turban, E., Lee, J., King, D., Chung, H.M., "ELECTRONIC COMMERCE- A Managerial Perspective," 台北：華泰，2000。
- [60] Van Heijst, G., Schreiber, A. Th., and Wielinga, B. J., "Using Explicit Ontologies in KBS Development," International journal of human and computer studies, Vol. 46, Iss. 2-3, 1997, pp.183-392.
- [61] Ward, J. and Aurum, A., "Knowledge Management in Software Engineering-Describing the Process," Software Engineering Conference, Australian 2004, pp.137-146.
- [62] Wiig, K.M., "Ont he Management of Knowledge," <http://www.km-forum.org>, 1999.
- [63] Yan, Y., Kuphal, T., and Bode, J., "Application of Multiagent Systems in Project Management," International Journal of Production Economics, Vol. 68, 2000, pp.185-197.
- [64] Zhuge, H., "Knowledge Flow Management for Distributed Team Software Development," Knowledge-Based Systems, Vol. 15, Iss. 8, 2002, pp.465-471.