

蔡文隆、陳仲儼、李明達 (2016),『應用本體論來協助軟體專案進行流程調適』,《中華民國資訊管理學報》,第二十三卷,第四期,頁 377-406。

應用本體論來協助軟體專案進行流程調適

蔡文隆*

亞東技術學院資訊管理學系

陳仲儼

國立中央大學資訊管理學系

李明達

國立中央大學資訊管理學系

摘要

對於軟體開發管理而言,組織通常會建立流程標準來讓各軟體專案據以執行。但鑒於軟體開發的獨特性,並非一套標準流程就可以完全套用在所有的軟體專案上。因此,企業組織需要訂定一套流程調適準則,讓專案成員根據不同專案所面臨的情境,依據調適準則而去調整實際的作法。然而,軟體流程調適是一個知識密集的活動,調適準則的制定或引用乃需要組織經驗方能提供適切的調適建議。故本研究嘗試建立一個以規則為基底的決策支援系統,運用本體論(Ontology)來建置一個知識與經驗導向的軟體流程本體模型,並據以發展出軟體專案流程調適推薦系統(Ontology-based software process tailoring recommendation system; OntoSPRS)。本系統透過推論規則的設計,收集調適經驗,並以推論引擎對知識本體進行推論並產生調適策略,可協助專案成員在軟體開發過程中能更完整地將組織經驗帶入而導出具體實作方式。本研究並以一國內物流公司雲端企業資源規劃系統之實際案例來展示本研究所建構的軟體流程本體。

關鍵詞：本體論、軟體專案、軟體流程調適、知識密集、決策支援

* 本文通訊作者。電子郵件信箱: wltsai@mail.oit.edu.tw
2014/11/25 投稿; 2015/6/26 修訂; 2015/09/08 接受

Tasi, W.L., Chen, C.Y. and Lika, I. (2016), 'Applying ontology to software process tailoring', *Journal of Information Management*, Vol. 23, No. 4, pp. 377-406.

Applying Ontology to Software Process Tailoring

Wen-Lung Tsai*

Department of Information Management, Oriental Institute of Technology

Chung-Yang Chen

Department of Information Management, National Central University

Irwan Lika

Department of Information Management, National Central University

Abstract

Purpose—Owing to the uniqueness in software development, this study develops an ontology-based software process tailoring recommendation system to manage quality and consistency in software development.

Design/methodology/approach—The system includes a knowledge- intensive software process ontology model and features a rule inference engine for deriving tailoring strategies.

Findings—This study found that the proposed system was able to improve tailoring recommendation outputs and make the underlying tailored strategy knowledge structure more accurate.

Research limitations/implications—Due to high project variability and complicated influence factors, more new research is needed in the future. Furthermore, the interactions among tailoring factors will derive even more tailoring rules in practice.

Practical implications—Since the system developed is specifically configured to work with the case company, the system may not be able to meet all organizational requirements. In practice, individual organizations should customize the content of

* Corresponding author. Email: wltsai@mail.oit.edu.tw
2014/11/25 received; 2015/06/26 revised; 2015/09/08 accepted

ontology, based on the organizational characteristics and its own process standard.

Originality/value— This study is the first develop a rule-based decision support system for software process tailoring. The study assists project specialists with project tailoring that complies with the organizational tailoring standard.

Keywords: Ontology, software project, process tailoring, knowledge-intensive, decision support

壹、緒論

現今軟體的發展與管理皆是以專案的方式來進行 (Chen et al. 2013)。軟體專案是一系列複雜的程序或流程，也是一個知識密集的過程，而成功關鍵之一取決於是否能善用組織的經驗 (Reich et al. 2012)。而透過專案的最佳實務來建立軟體流程模型，可成為組織持續學習的基礎，也可做為軟體工程活動管理的基準 (Reich et al. 2012)。因此，軟體組織通常會建立流程標準來讓各個專案據以執行，以統一管制所發展出來的產品品質，也用來累積並修正最佳做法與經驗。但鑒於軟體專案開發具有獨特性，在實務上並非一套標準流程或作法即可完全套用在所有的專案。若專案完全遵照一套開發標準而未依專案特性來加以調整，往往造成成本的無謂增加並徒增時間的浪費而導致專案的失敗 (Park et al. 2006; Kaur & Sengupta 2013)。

為了避免上述問題的發生，組織通常會針對軟體專案之特性，透過「調適 (Tailoring)」方式，調整軟體開發標準準則之流程定義與作法，進而成為一個新的流程供專案成員參考採用，此稱為「軟體流程調適 (Software process tailoring)」。(Ginsberg & Quinn 1995; Singh 1996)。在軟體流程調適的議題上，已有流程調適框架 (Demirors et al. 2000; Williams & Cockburn 2003) 之研究，但這對流程調適的工作是不足的，尚需領域的專業知識和經驗準則 (Park et al. 2006)，甚至是調適人員的個人能力經驗 (Clarke & O'Connor 2012; Martínez-Ruiz et al. 2012)、使用之系統工具 (Park et al. 2006; 陳仲儼 & 黃永福 2009; Alegria et al. 2014) 直接的影響了調適後的流程品質。例如，由於專案規模不大且開發時程較短，因而可簡化標準流程，又或者為了配合客戶及因應客戶要求而替換掉不同格式的相同產出。然而，若軟體流程調適不當，例如因為經驗與知識不足只好一味地遵從標準，以致於產生過多不必要的工作而增加額外的成本；反之或者流程過於被省略或刪除而沒有達成特定的工程與管理目標，也會導致無法保存專案知識 (Park & Bae 2013; Lorenz et al. 2014; Samuel et al. 2015)。如此高難度的軟體流程調適對於經驗與程度不一的各專案而言，是一大挑戰，其有賴於組織過去曾執行之專案所累積的知識與經驗來導引流程的調適。因此，如何以專案知識來引導個別專案進行調適和做出適宜的調適策略建議、以及如何協助專案人員有效率和有效益的進行流程調適，便成為軟體專案發展管理中的一個重要研究課題。

在知識活動中，本體論 (Ontology) 被用以整合並處理大量的知識 (Oh et al. 2011; Simperl et al. 2012)。本體論的作用是擷取一個特定的領域知識，針對該領域知識給予通用解釋，以提供使用者與電腦一個共通的詞彙與框架的定義，進而讓使用者對該領域知識的概念有一個共同理解。近年來本體論在資訊系統開發之

應用蓬勃發展，這是因為其能有效地將人類知識系統化，並透過共同理解性來達到知識分享的目的（Haghighi et al. 2013）。對於知識密集與經驗導向與高度再利用性的軟體專案管理領域來說，本體論是值得加以探討與嘗試應用。因此，本研究嘗試運用本體論於軟體流程調適，並開發出一個資訊系統，來協助組織在軟體專案開發過程中能適切地將專案知識（Xu & Ramesh 2008; Park & Bae 2013; Lorenz et al. 2014; Samuel et al. 2015）帶入而導出具體的實作方式。本研究將此系統命名為「軟體專案流程調適推薦系統（Ontology-based Software Process Tailoring Recommendation System）」，簡為 OntoSPRS。

在架構上，OntoSPRS 主要是由領域知識庫與推論引擎所組成。OntoSPRS 的知識庫是為收集組織的調適經驗，其基礎即是一個軟體流程知識本體模型；推論引擎則包含了知識規則的設計，以對知識本體進行推論並產生調適策略的推薦。在系統實作部份，OntoSPRS 應用本體開發編輯工具—Protégé，並透過 SWRL 規則語言來撰寫推論引擎。OntoSPRS 利用 Protégé 上的推論引擎來將本體與規則連結，並推論出相對應之推薦結果。此外 OntoSPRS 是以網頁為介面，以方便專案團隊能彈性地使用本系統來執行流程的調適。

為了展示研究的方法與結果，本研究也以一個實際專案為例，透過成員的實際使用來檢討所發展的系統之效益，並討論研究限制以及未來的研究與系統發展方向。本研究接下來的結構如下：第貳章為文獻探討，其包含了研究背景文獻的蒐集、專案知識及本體論相關研究的評論；第參章為研究方法；第肆章為系統案例的展示；第伍章為驗證與討論；最後的第陸章則是文章總結以及未來發展方向。

貳、文獻探討

一、軟體專案特性與流程調適

雖然組織為方便軟體專案的品質與管理而建立流程標準。但因專案具有獨特性，同一套流程標準並非能夠完全套用於所有專案中。然而，若因每次建置新專案而重新設計流程，將會徒增組織的資源支出，也對專案成效的助益不大（Xu & Ramesh 2007; Alegria et al. 2014）。因此，新建之專案應依組織現有流程或標準並針對專案特性進行調適，以滿足並達到新建專案的需求與目標。因此，許多知名的流程標準參考模式也都強調軟體流程調適的重要性。例如，能力成熟度整合模式將流程調適利用多個流程領域（如 OPF 與 IPM）來定義流程調適相關規範，並以成熟度三級來明確要求組織必須有能力進行流程的調適（CMMI Product Team 2010）。

專案在進行流程調適前，需透過專案的特徵找出影響調適的因素（Martínez-

Ruiz et al. 2012)。專案特徵包括技術複雜度、協同發展、客戶配合度、流程引用、專案型態、專案時程、專案規模、預算、團隊規模、軟體生命週期、軟體型態、開發模式、團隊能力度、需求明確程度等 (Ginsberg & Quinn 1995; Nidumolu 1996; Park et al. 2006; Na et al. 2007; Xu & Ramesh 2007; Chen & Tsai 2011; Liu et al. 2011; Martínez-Ruiz et al. 2012; Kalus & Kuhrmann 2013; Sulayman et al. 2014)。本研究透過大規模的文獻整理來探討並歸納廣泛討論與應用的流程調適因子，可被調適的流程物件主要包含了流程活動 (Activities)、文件 (Documents)、產出 (Artifacts)、任務角色 (Roles)、階段 (Phases) 等，其屬性包括流程物件正式程度的形式 (Formality)、流程活動被執行的頻率 (Frequency)、流程物件細節程度的顆粒度 (Granularity) 與流程所涵蓋的範圍 (Scope)。

由上而知，軟體開發所涉及層面相當錯綜複雜，故 Aaen (2008)、Ebert and De Man (2008) 則將軟體專案分成產品 (Product)、專案 (Project)、流程 (Process) 及人員 (People) 等四大構面，這也是軟體工程領域所稱的 4P，這四個構面包括所有影響專案成敗的因素，包括調適因素在內，這四個構面相互關連、牽扯與影響，例如：專案規模小且開發時程短，可將流程活動和工作成品予以豁免或使用較不正式的形式，可將產出的計畫書改為計畫單；或者變更相關人員所扮演的角色，如人員不夠時，專案經理可擔任測試人員。

軟體流程調適自 ISO/IEC 12207:1995 (Singh 1996)，Ginsberg and Quinn (1995) 提出後，大多僅提供流程指引與調適框架 (Demirors et al. 2000; Williams & Cockburn 2003) 之研究，但這對流程調適的工作是不足的，尚需領域的專業知識和經驗準則 (Park et al. 2006)，甚至是調適人員的個人能力經驗 (Clarke & O'Connor 2012; Martínez-Ruiz et al. 2012)。流程調適可使用輔助工具模型協助流程活動之調適 (Alegria et al. 2014)，Park et al. (2006)、陳仲儼與黃永福 (2009) 分別應用類神經網路、模糊理論發展出一套決策支援系統，協助專案進行調適。這三個研究皆應用了組織經驗的累積並將之編撰成調適規則。調適規則通常是廣泛且獨立於特定專案情境的廣義知識 (generalized knowledge)。組織雖然可以透過調適規則來產生調適建議，但在實際的情況下，專案人員還需要經驗案例以應用或套用所產生出來的規則或作法。經驗案例是一種情境化的知識 (contextualized knowledge)，其描述了專案當下的具體情境和其所採取的決策及結果。Park et al. (2006)、陳仲儼與黃永福 (2009)、Alegria et al. (2014) 的研究專注在廣義知識的建立，而本研究更進一步地把情境化知識納入於所設計之系統 (OntoSPRS) 中，透過情境化知識的輔助以協助專案進行調適。

二、專案知識

專案發展過程中，專案人員大都會使用現有組織的背景知識與資源，並經分析使用後需產生新的知識，可作為組織往後的專案知識基礎。在專案管理背景知識中，分成上述的廣義知識和情境化知識 (Xu & Ramesh 2008; Park & Bae 2013; Lorenz et al. 2014; Samuel et al. 2015)。

廣義知識係指能明確表達的顯性知識，例如：組織規定或包含在工作準則內的作業程序與說明。一般來說，組織訂定規則或工作準則內容，大都來自一些相關的流程規範或是標準，如能力成熟度模型整合 (Capability Maturity Model Integration; CMMI)、專案管理知識體系 (Project Management Body of Knowledge; PMBOK)、ISO/IEC 12207 (Software Life Cycle Processes)、ISO/IEC 15504 (Software Process Improvement and Capability Determination) 等。這些流程規範或標準，定義了專案生命週期 (Life Cycle)、角色 (Role)、作法 (Operation)、作業程序 (Procedure) 及產出 (Output) 等專案顯性知識 (陳仲儼 & 黃永福 2009)。

情境化知識則是指知識產生的途徑及應用情境，例如專案在面對各種情境變通之產品、流程及人員等專案調適因子 (Project factor)、因應措施之調適作業 (Tailoring operation) 及調適決策 (Tailoring strategy) 結果。在應用上，情境化知識可供決策者做為決策的參考依據，例如，中小型專案進行調適時，專案人員可參考並依據過去的中小型專案之調適經驗來進行調適；也就是說情境化知識包括了組織過去曾執行過專案執行紀錄、流程的調適作法、產出調適建議等個人經驗，以及參考過往其它專案資訊後，所轉化成個人的知識，都是情境化知識。

為協助專案人員做出正確的決策，許多研究 (Park et al. 2006; 陳仲儼 & 黃永福 2009; Alegria et al. 2014) 指出須有一知識管理系統 (Knowledge-based system, KBS) 輔助，例如決策支援系統，除能提供決策建議外，亦可知決策建議產生之流程。因過去的決策之記載或經驗可提供專案人員所欠缺的相關特定知識，以提升專案人員對 KBS 邏輯的理解，故在應用廣義知識的同時，應再納入情境化知識供專案人員參考以做出符合當下情境的決策。流程調適工作大都是在沒有預定的變通方法的新情況下進行，專案人員需依據情境的範圍而做出合理的判斷和決策。對資深專案成員而言，可輕易將專案現況轉化成情境化知識，促進知識的轉移和重複使用；對於經驗不足專案成員而言，則有賴組織所保留的過往專案及成員轉化的情境化知識，降低調適不當情形發生；相對地，若組織未能完整保留情境案例，也就是歷史專案相關資訊，所獲得的知識片面不全，便僅能依廣義知識之建議，無法適切的引用調適規則或把調適建議套用在問題上。

三、本體論

本體論主要是用來整合並處理大量的知識，其作用是擷取一個特定的領域知識，針對該領域知識給予通用解釋，以提供使用者與電腦一個共通的詞彙與框架的定義，進而讓使用者對該領域知識的概念有一個共同理解 (Chi 2010)。這對於知識密集與經驗導向與高度再利用性的軟體專案來說，本體論是值得加以探討與應用。在軟體組織中，為了能有效整合過往的專案知識並使之成為組織經驗，需對專案知識進行塑模 (Simperl et al. 2012)。然而，軟體開發過程中需要工具來協助擷取和表達專案知識，以便能呈現專案執行情形和所須知識獲取。而本體論被應用於許多領域，主要是因為有可以明確地描述語意和關係的電腦語言 (例如：XML) 之表達能力，也就是說本體論提供專案成員與電腦一個共通的詞彙與框架的定義，進而讓專案成員對軟體流程知識的概念有一個共同理解。另外，本體論除可系統化地闡明軟體流程領域概念外，本研究將概念結構進行塑模以建置軟體流程的本體模型，並進一步進行分析與推論，用以回饋專案成員，達到專案知識共享之目的 (Simperl et al. 2012)。

在將本體論系統化的過程中，OWL (Web Ontology Language) 是用來描述本體論知識的一種標準；OWL 是以 XML 為基礎，提供多樣性的詞彙，且能明確表示辭彙或概念之間的關係，使系統更容易處理與分析語意內容。OWL 主要用途在表達本體上層之概念架構，而對於本體下層的實例之間關係則可搭配 SWRL (Semantic Web Rule Language)，將整個本體之上下層緊密串連 (Chi & Chen 2009)。SWRL 為了 OWL 所發展出來的規則語言，透過 SWRL 產出的規則可輕易的與 OWL 知識庫連結，因此 SWRL 可直接使用 OWL 定義好的詞彙與關係，而不必另外或自行定義，亦可減少衝突或錯亂。另外，透過 SWRL 規則可從 OWL 知識庫中推論出新的知識，這對於知識密集的軟體流程調適來說，是極具重要的工具與產出。因此，在文中以 OWL 來描述與建置知識本體架構並輔以 SWRL 進行實例之間的關係描述與推論規則應用，藉此建構本研究軟體專案流程調適完整之本體知識模型。

參、系統設計

一、系統架構

在專案發展的過程中，專案人員雖然可依據流程調適準則來進行流程的調適，但在實際的情況下，調適準則的制定與引用則是需要知識與經驗的累積。因此，本研究提出一個知識與經驗導向、調適專案流程並進而功能化的架構。軟體專案在專案章程確立的同時，可透過一架構所建的軟體流程知識本體來協助經驗

不一的專案成員順利調適流程、並給予實作上之推薦。

本研究的系統架構包括軟體流程知識本體、推論引擎、調適策略的知識架構以及使用者介面四個部分（如圖 1 所示）。首先，就實作而言，本研究將以 OWL 來描述軟體流程的本體知識並應用 Protégé 來建立其本體模型；推論規則是以 SWRL 來撰寫並利用 Protégé 所提供的 HermiT 推論引擎來將 SWRL 規則進行推論並產生調適策略的知識架構。本研究之決策支援系統將以此知識架構為基礎，透過使用者介面與使用者進行溝通並提供調適策略的建議。

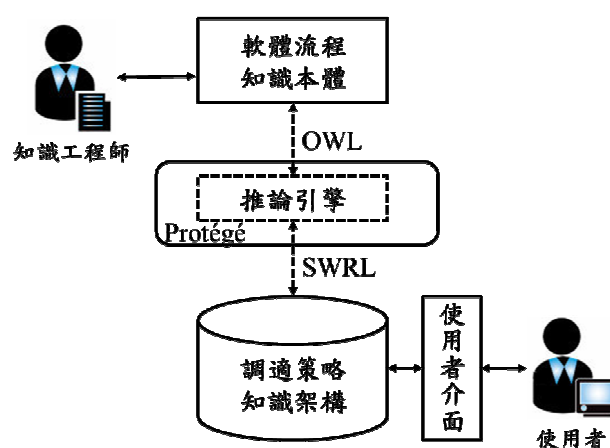


圖 1：系統架構圖

其次，說明此系統之關鍵人員。對軟體流程知識本體而言，是由知識工程師所建立並維護。在實務上，知識工程師可以是具豐富軟體專案經驗之人員或者由組織的軟體流程改善小組（Software Engineering Process Group; SEPG）（CMMI Product Team 2010）的成員，且知識工程師可以群體方式共商完成，或經由專案領域專家的協助，以確保其內容的正確性。再者，系統架構包含使用者介面的開發，並透過這些介面來供專案經理或其成員（i.e. 使用者）與系統進行溝通，並查詢所執行的專案所需求的流程調適之策略與建議（即圖中的調適策略知識架構）。

根據前一章的文獻探討，軟體工程領域的 4P—產品、專案、流程及人員四大構面包含所有軟體專案調適因素（Aaen 2008; Ebert & De Man 2008），這些因素也就是專案廣義知識，故本研究將之作為軟體流程本體中的主（父）概念；另外，透過相關文獻的整理，本研究歸納出十五項專案調適因子如表 1 所示，本研究即採用這十五項調適因子作為四大因素之子概念，本研究之調適概念本體即以此為依據，並延伸建置分類實例，表 1 即是此流程調適概念集合的定義，而概念內容

則有賴於企業實例的收集。

表 1：本研究的專案流程調適概念集合

調適類別	調適子類別	資料來源
產品因素	產品規模	Nidumolu 1996; Park et al. 2006; Na et al. 2007; Prikładnicki et al. 2007; Xu & Ramesh 2007; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	產品型態	Xu & Ramesh 2007; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	需求明確程度	Nidumolu, 1996; Na et al. 2007; Xu & Ramesh 2007; Liu et al. 2011; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	需求變動程度	Nidumolu, 1996; Na et al. 2007; Xu & Ramesh 2007; Liu et al. 2011; Kalus & Kuhrmann 2013; Sulayman et al. 2014
專案因素	專案型態	Ginsberg & Quinn, 1995; Nidumolu, 1996; Park et al. 2006; Na et al. 2007; Xu & Ramesh 2007; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	專案時程	Nidumolu, 1996; Na et al. 2007; Xu & Ramesh 2007; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	專案預算	Nidumolu, 1996; Na et al. 2007; Xu & Ramesh 2007; Kalus & Kuhrmann 2013; Sulayman et al. 2014
流程因素	承包模式	Park et al. 2006; Prikładnicki et al. 2007; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	協同發展	Ginsberg & Quinn, 1995; Park et al. 2006; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	流程引用	Kalus & Kuhrmann 2013; Sulayman et al. 2014
人員因素	人員能力度	Ginsberg & Quinn, 1995; Nidumolu, 1996; Park et al. 2006; Na et al. 2007; Xu & Ramesh 2007; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	供應商配合程度	Yan & Dooley 2013; Yan & Dooley 2014
	客戶配合度	Nidumolu, 1996; Park et al. 2006; Xu & Ramesh 2007; Liu et al. 2011; Kalus & Kuhrmann 2013; Sulayman et al. 2014
	高階管理者支持	Xu & Ramesh 2007; Elbanna 2013; Kalus & Kuhrmann 2013; Young & Poon 2013
	專案團隊規模	Park et al. 2006; Prikładnicki et al. 2007; Xu & Ramesh 2007; Kalus & Kuhrmann 2013; Sulayman et al. 2014

二、軟體流程本體之建立

本研究使用 Protégé 做為本體編輯和知識獲取的工具，建立其知識本體，再根據 Chi (2010) 做法並且將知識本體分成兩個主要的類別：

- (1) 領域本體 (Domain ontology)，用以描述軟體專案中的廣義知識。本研究所設計「領域本體」(圖 3) 中，包含在第貳章文獻探討中，所提及軟體專案特性之「專案調適因子」，與流程規範（例如：CMMI、PMBOK）或標準（例如：ISO/IEC 12207），定義之「調適策略」、「調適作業」、以及「流程物件」等軟體流程調適領域的廣義知識，因此將之歸類為領域本體之類別。而「專案調適因子」及「流程物件」類別則可進一步分類出第二層架構的子類別 (Subclass)。本研究透過這些類別實例 (Instance) 的建立，組成領域本體之知識框架。
- (2) 任務本體 (Task ontology)，用以描述軟體專案中解決方案之概念，而任務是由廣義知識之間（例如：「專案調適因子」、「調適策略」）的關聯所衍生之新類別，也就是情境化知識。在圖 4 中，本研究將軟體流程調適廣義知識之關聯，提出「因子分析」(FactorAnalysis)、「調適作法」(TailoringDecision)、「流程調適知識」(TailoringKnowledge) 以及「調適建議」(TailoringRecommendation) 等四個類別。而「調適作法」類別對應至領域本體中的「流程物件」項下子類別，則可再進一步分類出第二層架構的子類別。

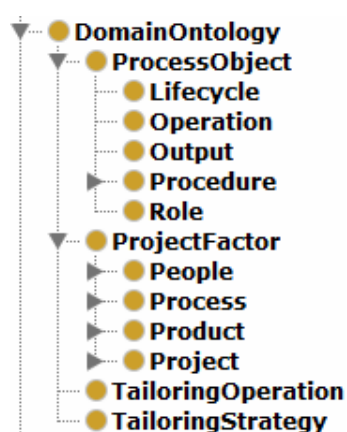


圖 3：領域本體類別階層

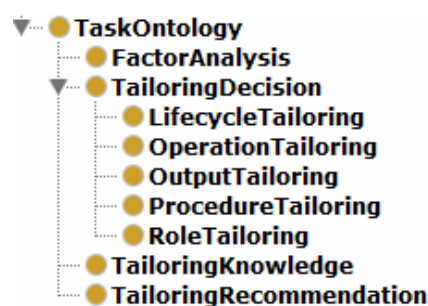


圖 4：任務本體類別階層

任務本體也定義了廣義知識與概念之間的關係框架，其利用屬性來建立概念之間的關係，通常以 has-a 或 part-of 屬性來表達一項概念擁有其他某項概念。以本研究「調適建議」類別為例（表 2），來說明其內部組成。「調適建議」類別共設計六項屬性，其中 hasFactor 記錄了「調適建議」所對應的「專案調適因子」。其他五項屬性均為推論形式，亦即內容值未知的屬性，須透過已知的事實和推論規則，分別推論出該調適因子的「生命週期調適建議」（hasLifecycleTailoringRec）、「流程作業調適建議」（hasOperationTailoringRec）、「作業程序調適建議」（hasOperationTailoringRec）、「流程產出調適建議」（hasOutputTailoringRec）以及「人員角色調適建議」（hasRoleTailoringRec）。

表 2：「調適建議」類別之內部組成

屬性	類別	宣告／推論	值域
hasFactor	Object	asserted	ProjectFactor
hasTailoringStrategy	Object	inferred	TailoringStrategy
hasLifecycleTailoringRec	Object	inferred	LifecycleTailoring
hasOperationTailoringRec	Object	inferred	OperationTailoring
hasProcedureTailoringRec	Object	inferred	ProcedureTailoring
hasOutputTailoringRec	Object	inferred	OutputTailoring
hasRoleTailoringRec	Object	inferred	RoleTailoring

為了避免知識本體所涉及的範圍因為太過於廣泛而變得太複雜，因此在知識本體建置前，必須先界定知識本體的領域範圍，再將軟體流程做為主要的知識框架並專注於流程調適作法的推導及隱含知識的擷取。本研究根據所設定的知識框架並且進一步進行概念知識的收集，透過所收集的概念知識來進行塑模並建立其知識本體，如圖 2 所示。

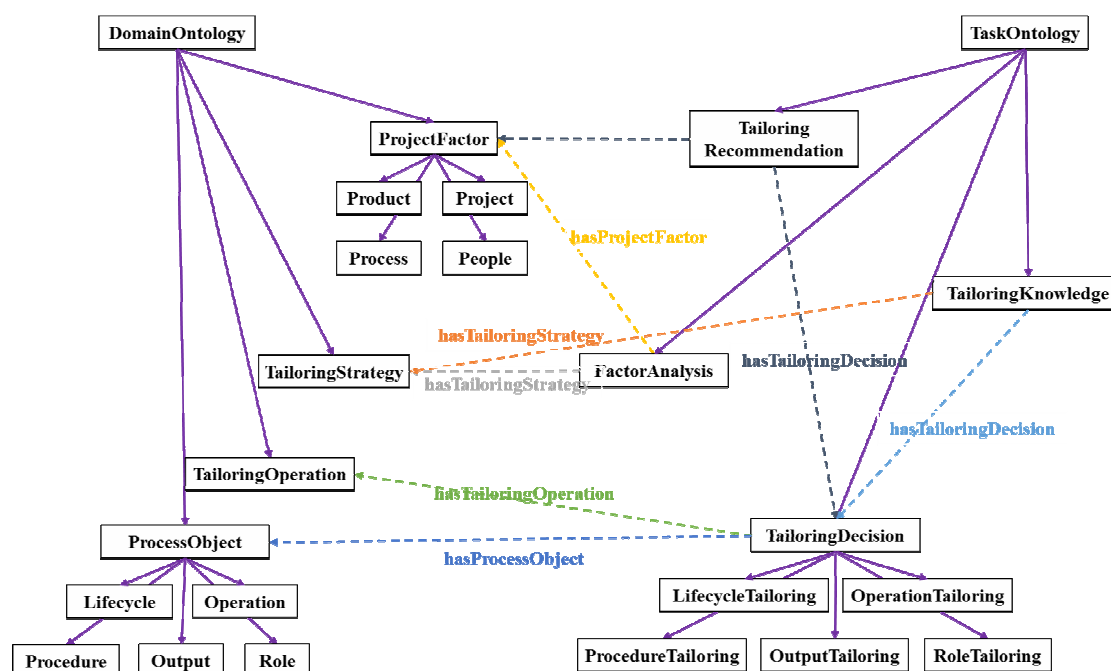


圖 2：軟體流程本體模型

三、規則設計

在完成知識的塑模與建置之後，即可進行推論規則的設計。前述任務本體的建置即是為了推論機制的運用。任務本體中，推論值（Inferred property）之屬性即代表該屬性之內容值為未知的隱性知識。透過推論規則的設計，針對知識本體中所描述的關係來進行推論以找到隱藏於所定義好的關係中之隱性知識。以表 2 為例，調適策略屬性（hasTailoringStrategy）以及生命週期調適建議屬性（hasLifecycleTailoringRec）皆為內容值未知的屬性。首先，目標專案需藉由「專案調適因子」以推導出其「調適策略」，接著再透過先前所推導出的「調適策略」來對「生命週期調適建議」的建議作法進行推論。

在建立任務程序的過程中，通常會先鑑定出問題及其解決方案。本研究所欲解決的問題即軟體流程調適的建議作法，也就是前述所提及的推論值。在規則設計的過程中，推論值即為規則的結論（Consequent），而規則的建立則將著重於規則前提（Antecedent）之設計。本研究根據 Chi（2010）所提的規則分析表來進行解題分析及推論規則的設計。此方法以自然語言進行陳述，並以 $km\{Goal\ (inferred\ property) : step_1, step_2, \dots, step_n\}$ 的格式來表示，模擬如何透過已知的知識之間的關聯做連結，並逐步進行分析，直到得到結論為止。

針對推論規則的撰寫，目前已發展出許多種規則語言，例如 RuleML、SWRL、Metalog、ISO Prolog 等。其中，SWRL 則是針對語意網的應用所特別制定的，其語法簡單易懂，且可輕易地與 OWL 知識庫做連結。SWRL 規則可用來提供程序性知識 (Procedural knowledge) (Chi 2010)。程序性知識是一種需透過某種作業形式而間接推論出來的知識。其可用以彌補本體論的推論功能上的一些限制，尤其在個體之間的語意關係的識別上 (Horrocks et al. 2005)。「前提 (Antecedent) → 結論 (Consequent)」為 SWRL 的典型邏輯表示式，用以表示其語意規則 (Semantic rule)。前提通常由多個單元 (atom) 所串聯而成，因此可以將其規則形式寫成「 $(atom_1, atom_2, \dots, atom_n) \rightarrow consequence$ 」的格式。此處的 atom 即為規則分析表中的 step。依照 SWRL 的規則形式，規則分析表中的 goal 即為規則的結論，也就是須經由規則推論而獲取的推論值。本研究將專案的「調適策略」和「調適建議」的推論規則歸納如表 3 所示。Rule 1 規則分析表中的「調適策略」之推論規則，亦即「專案調適因子」的「因子分析」，用以推導「調適建議」分別進一步發展 Rule 2「軟體生命週期調適建議」、Rule 3「流程作業調適建議」、Rule 4「作業程序調適建議」、Rule 5「流程產出調適建議」以及 Rule 6「人員角色調適建議」之推論。

表 3：SWRL 推論規則

Rule 1	TailoringRecommendation(?x), TailoringRecommendation_hasFactor(?x, ?y), FactorAnalysis(?z), FactorAnalysis_hasFactor(?z, ?y), FactorAnalysis_hasTailoringStrategy(?z, ?a)→ TailoringRecommendation_hasTailoringStrategy(?x, ?a)
Rule 2	TailoringRecommendation(?x), TailoringRecommendation_hasTailoringStrategy(?x, ?y), TailoringKnowledge(?z), TailoringKnowledge_hasTailoringStrategy(?z, ?y), hasLifecycleTailoring(?z, ?a)→ hasLifecycleTailoringRec(?x, ?a)
Rule 3	TailoringRecommendation(?x), TailoringRecommendation_hasTailoringStrategy(?x, ?y), TailoringKnowledge(?z), TailoringKnowledge_hasTailoringStrategy(?z, ?y), hasOperationTailoring(?z, ?a)→ hasOperationTailoringRec(?x, ?a)
Rule 4	TailoringRecommendation(?x), TailoringRecommendation_hasTailoringStrategy(?x, ?y), TailoringKnowledge(?z), TailoringKnowledge_hasTailoringStrategy(?z, ?y), hasProcedureTailoring(?z, ?a)→ hasProcedureTailoringRec(?x, ?a)
Rule 5	TailoringRecommendation(?x), TailoringRecommendation_hasTailoringStrategy(?x, ?y), TailoringKnowledge(?z), TailoringKnowledge_hasTailoringStrategy(?z, ?y), hasOutputTailoring(?z, ?a)→ hasOutputTailoringRec(?x, ?a)
Rule 6	TailoringRecommendation(?x), TailoringRecommendation_hasTailoringStrategy(?x, ?y), TailoringKnowledge(?z), TailoringKnowledge_hasTailoringStrategy(?z, ?y), hasRoleTailoring(?z, ?a)→ hasRoleTailoringRec(?x, ?a)

肆、系統與案例展示

一、個案描述

隨著雲端應用的蓬勃發展，企業資訊系統以雲端架構整合企業內外部各項作業與資源成為了主流的趨勢，H 公司系統開發亦朝此方向發展。H 公司成立於 1938 年，其發展過程由傳統的運輸公司轉型為現代化的資訊服務業，提供物流、商流、金流以及資訊流整合之綜合型物流服務。本章以 H 公司之「Cloud ERP」專案展示本體實例與系統展示案例。

在 H 公司過去의專案經驗中，常因調適不周導致專案包含過多不必要的流程及文件工作上，這因為該公司專案成員缺乏調適經驗所造成的情況。H 公司往往在付出訓練成本後，人員即因有了經驗而轉任其他企業。因此，如何將人員知識與經驗有系統地保留並傳承，成了 H 公司所重視的課題，即需要建構一知識模型來將人員的知識與經驗保留下來並妥善維護，以便給專案成員做參考及使用。為使專案成員在專案的進行中做調適建議的查詢，故將 H 公司「Cloud ERP」之「需求發展與系統分析設計」流程作業之實例（如表 4），對應本研究表 1 歸納產品規模等十五項專案調適因子，進行功能說明與展示操作，冀能在完成實例的新增後，進行規則的推論，藉由推論引擎來推導出針對目標專案調適因子或不同調適因子之組合的調適建議，並將之寫回本體之中。

表 4：「Cloud ERP」專案實例

調適因子之分類	專案調適因子	實例
產品因素	產品規模	6000 個功能點
	產品形態	ERP
	需求明確程度	明確
	需求變動程度	低
專案因素	專案型態	開發型專案
	專案時程	5 個月
	專案預算	NT\$50 萬
流程因素	承包模式	在岸／境內內包
	協同發展	否
	流程引用	否
人員因素	人員能力度	普通
	供應商配合程度	高

	客戶配合度	高
	高階管理者的支持	支持
	專案團隊規模	20 人

二、系統實作

在第參章完成軟體流程本體模型後，本研究據以建置成系統，並命名為 ONTOSPRS，圖 5 為 ONTOSPRS 的系統運作流程圖。假如目標專案之開發為一嶄新且初次建置之功能，企業可將此目標功能新增至知識庫中；若建置功能實例已於歷史專案中建置過並且存在於規則庫中，可透過查詢得目標專案之流程調適，於定義好此新的目標專案之發展工作後，系統後端會自行依照推論規則找到其流程調適實作建議。另外，若專案實例專案執行環境變動而需要進行調適項目的實作更新時，亦可透過系統的修改功能來達成。除了本體模型建置與系統實作外，ONTOSPRS 也提供 web-based 使用者介面（如圖 6），可讓專案成員收到專案啟動計畫書時，在任何地方可以連線到此系統，並透過此系統提供目標專案之流程類別查詢，將所對應的流程調適項目導出，或是得以在專案執行相關會議現場針對流程調適實作進行即時的討論與確認。

本研究利用 Protégé 來編輯本體模型內容，這部份即是參照第三章的領域本體（如圖 2）及任務本體（如圖 3）來進行。圖 7 為 Protégé 的編輯視窗之畫面，左邊方框是本研究所設計的軟體流程本體模型之類別階層（class hierarchy），中間方框則是新增實例之工作區，右上方框為 SWRL 規則的編輯視窗。

完成上述本體架構及系統後，開始進行各項軟體流程調適，以圖 7 內的 OperationTailoring_Exempt_RequirementDevelopmentAndSystemAnalysisAndDesign 的流程作業調適為例，透過中間方框的新增實例之工作區來新增此實例並透過右下方框的實例描述視窗將實例的類型或實例所屬的類別定義為流程作業調適（OperationTailoring）類別。透過左邊方框的類別階層視窗，亦可得知「流程作業調適」即為調適作法（TailoringDecision）的子類別之一。這即代表「簡化需求發展與系統分析設計」即為「流程作業調適」類別的一實例，同時也是「調適作法」的實例。右下方框的屬性編輯視窗是用以定義該實例的屬性之值域。「流程作業調適」則擁有兩個屬性，即 OperationTailoring_hasTailoringOperation 和 Operation Tailoring_hasOperation。以此例而言，其屬性分別對應到調適作業（TailoringOperation）類別的豁免（Exempt）以及需求發展與系統分析設計作業（RequirementDevelopmentAndSystemAnalysisAndDesign）的實例。由此亦可看出，「流程作業調適」類別便是透過「調適作業」及「流程作業」概念之間的關係而形成的。在完成實例的新增後，進行規則的推論，藉由推論引擎來推導出針

對目標專案調適因子或不同調適因子之組合的調適建議，並將之寫回本體之中。

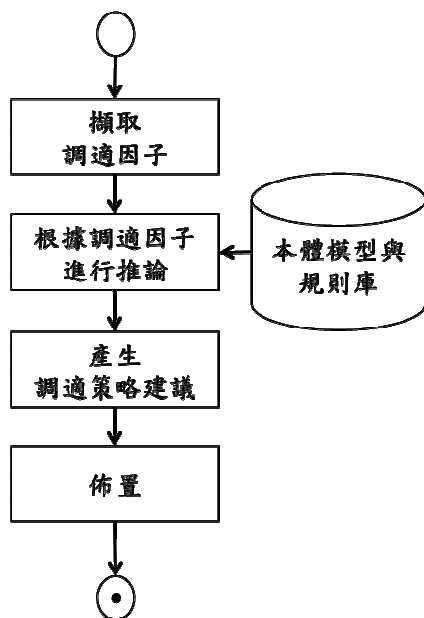


圖 5：ONTOSPRS 系統流程之活動圖

圖 6：調適因子擷取介面

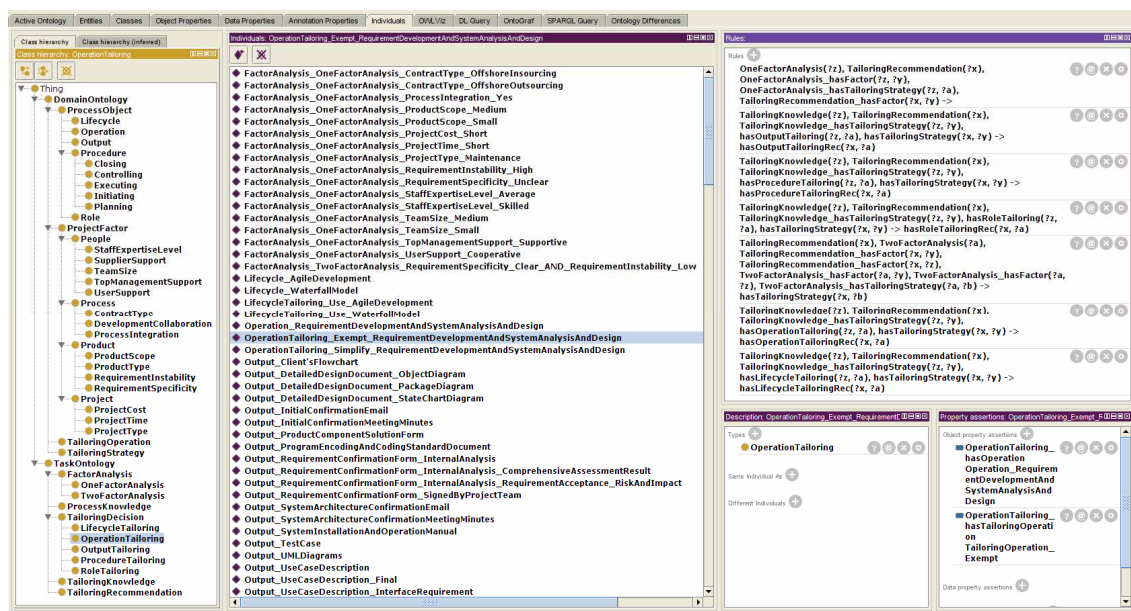


圖 7：Protégé 之編輯視窗畫面

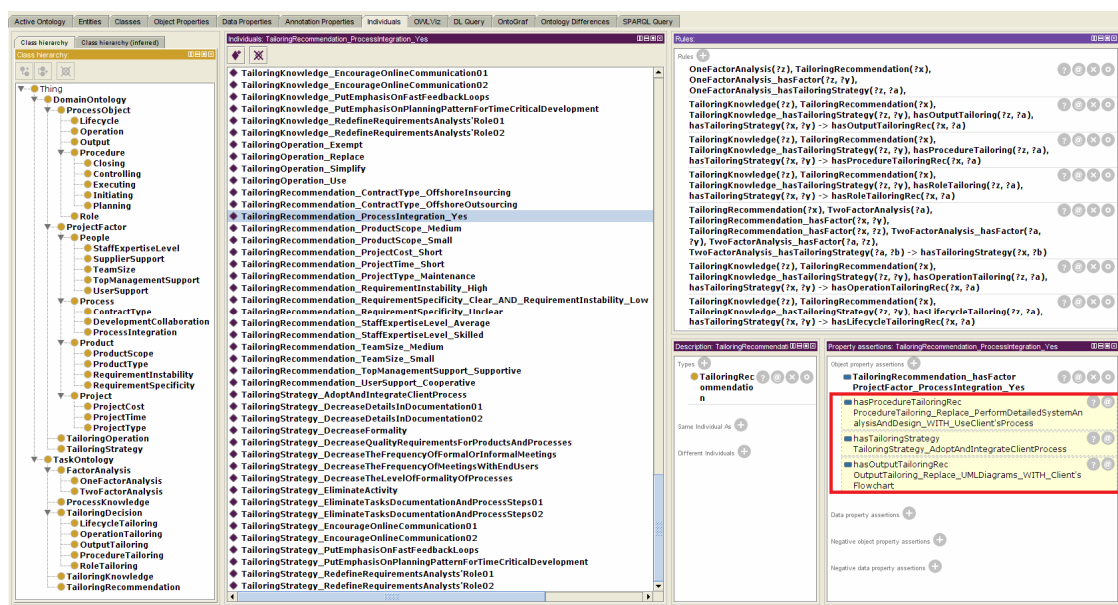


圖 8：「引用客戶流程」之「調適建議」

在啟動推論引擎之後，系統即可自動找出間接與隱藏的知識。例如圖 8 內的調適建議（TailoringRecommendation）類別的引用客戶，經由規則的推論後，右下方框的屬性編輯視窗便多出 3 項屬性，分別是 hasTailoringStrategy、hasProcedureTailoringRec 以及 hasOutputTailoringRec。在推論過程中，首先會先針對表 3 中 rule 1 的調適策略（TailoringStrategy）來進行推論。以本例而言，其「調適策略」的推論結果為導入並整合客戶流程（TailoringStrategy_AdoptAndIntegrateClient's Process）。在獲得「調適策略」的結果後，接下來便可透過表 3 中的 rule 2 - rule 6 來進行調適作法（TailoringDecision）的推論。在本例子中，其共推論出兩項「調適作法」，分別是作業程序調適（ProcedureTailoring）以及流程產出調適（OutputTailoring）。針對「引用客戶流程」這項「專案調適因子」，如圖 7 所示，細部系統分析與設計取代為採取客戶端流程（ProcedureTailoring_Replace_PerformDetailedSystemAnalysisAndDesign_WITH_Us eClientsProcess）以及將 UML 圖取代為客戶端流程圖（OutputTailoring_Replace_UMLDiagrams_WITH_Client'sFlowchart）兩種調適建議。其他「專案調適因子」皆按照與此方法來推論出其「調適建議」，最後產生調適策略的知識架構。本研究之決策支援系統將以此知識架構為基礎，透過使用者介面與使用者進行溝通並提供調適策略的建議。

三、系統之使用者介面

為便利使用者（H 公司專案成員）於進行軟體專案開發時查詢流程調適實作推薦，本研究之 OntoSPRS 系統界面，其主要的系統操作功能為調適建議的產生。

為了展示現有知識流程調適之建議，本研究續以前述「Cloud ERP」專案為例。當專案要開始進行調適時，系統將先從第一階段的調適因子之擷取來開始進行，專案成員可透過圖 6 中的調適因子擷取介面來輸入以上實例中的專案調適因子。本系統利用引導的方式，將調適因子的擷取分成四個層次來進行。如圖 6 所示，使用者首先輸入專案名稱，接著再依序點擊調適因子分類之標題以展開其調適因子的輸入畫面。在完成調適因子之輸入後，專案成員即可按下「調適」按鈕以進行調適因子，經本體的分析後，產生專案的調適建議，如圖 9 所示。

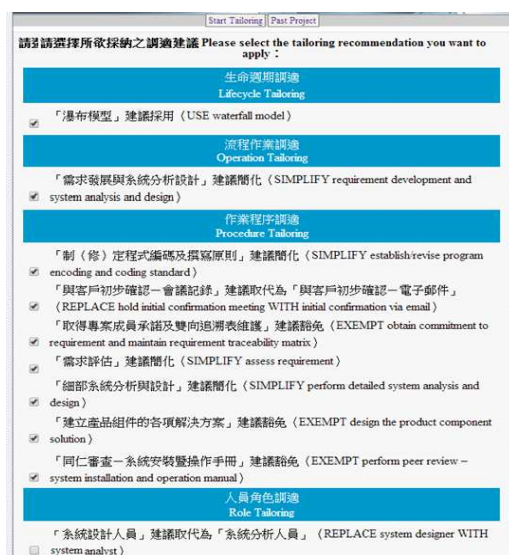


圖 9：調適建議擷取畫面

專案名稱 Project Name	日期 Date	專案型態 Project Type	檢視 View	再調適 Retailor	回饋 Feedback
Cloud ERP	2014-05-31	development 開發型專案	檢視 View	再調適 Retailor	回饋 Feedback
eLCO 電子商務平台	2014-05-21	development 開發型專案	檢視 View	再調適 Retailor	回饋 Feedback
IBIS 企業整合管理系統	2014-05-11	development 開發型專案	檢視 View	再調適 Retailor	回饋 Feedback
ITS 整合管理系統維護	2014-01-10	maintenance 維護型專案	檢視 View	再調適 Retailor	回饋 Feedback
OPUS 線上採購系統維護	2013-11-05	maintenance 維護型專案	檢視 View	再調適 Retailor	回饋 Feedback
OAST 線上競標系統	2013-09-03	development 開發型專案	檢視 View	再調適 Retailor	回饋 Feedback
OPUS 線上採購系統	2012-08-02	development 開發型專案	檢視 View	再調適 Retailor	回饋 Feedback
QeST 線上報價系統	2012-06-12	development 開發型專案	檢視 View	再調適 Retailor	回饋 Feedback
ITS 整合管理系統	2011-10-08	development 開發型專案	檢視 View	再調適 Retailor	回饋 Feedback

圖 10：歷史專案總覽擷取畫面

完成調適建議並專案成員確認之後，系統便可將此專案及其調適結果一併存到資料庫之中以供未來的專案做參考或進行再調適。儲存完專案的調適結果後，系統將跳至歷史專案的總覽頁面並將過去所有情境案例，也就是調適過的專案列出來，如圖 10 所示。「Cloud ERP」專案已被存進系統內，成一情境案例。專案人員可「再調適」功能來進行專案的再調適。另外，專案成員可透過「檢視」功能，參考它情境案例（例如：電子商務平台等歷史專案）之調適因子與調適建議；也就是說藉由系統的檢視功能可獲取到組織過去曾執行過專案知識，包括該專案成員針對其專案特性與情境、於流程上的調適作法以及產出調適建議等個人

經驗或知識。即便這些成員已離職，組織其它專案團隊可依系統所記錄的經驗與知識，以避免相似專案程序發生流程調適不周的情況。

伍、分析與討論

一、系統使用分析

對於新系統或非量化模型的使用情形之探討，在高度客製化及人本（Human centric）的使用環境來說，質性驗證是一可行的方式（Seaman 1999; Darke et al. 1998）。本研究設計之系統—OntoSPRS 實作於個案公司，其系統具高度客製特性，且使用者介面呈現之知識以人為本的訴求，故本研究以質性方式於個案公司人員使用情形進行分析與探討。本研究採用科技接受模式（Technology Acceptance Model: TAM）（Venkatesh & Davis 2000），以「認知有用（Perceived usefulness）」、「認知易用（Perceived ease of use）」、「主觀規範（Subjective norm）」與「使用意圖（Intention to use）」來探討本研究之本體應用系統之實用價值。

由於探討軟體工程相關問題之質性研究中，半結構式訪談（Semi-structured interview）普遍被運用（Myers & Newman 2007）。因此本研究使用半結構式訪談來蒐集資料，受訪對象則設定為 OntoSPRS 的目標使用者，即 H 公司的專案成員，亦即軟體開發工程師（A 與 B）、資訊處副理與營運長。表 5 為訪談對象的基本資料與工作內容，這些受訪對象分別屬不同階層（專案執行、管理階層與營運階層），亦藉由不同業務角度來共同檢視系統的有用性與易用性，也讓讀者能更清楚掌握訪談對象與訪談內容的關聯性、組織結構（即人員的層級與業務角色）與面向（例如，從專案執行、組織對於相關專業知識的保存與掌握、到對企業永續營運的影響）。

表 5：訪談對象的基本資料與工作內容

人員	相關學歷	資訊年資	專案工作
營運長 （高階經理）	資訊管理科系 大學畢業	20 年以上	協助爭取專案與立案、專案里程碑管理、協調專案資源
資訊處副理 （專案經理）	資訊管理科系 大學畢業	15 年以上	負責統籌與規劃公司的專案、客戶與需求訪談、系統架構規劃、流程調適
軟體工程師 A	資訊工程科系 大學畢業	5 年	負責執行客戶與需求訪談、系統分析設計、程式開發
軟體工程師 B	資訊工程科系 研究所畢業	2 年	負責協助系統分析設計、程式開發與測試、專案內其他行政工作

（一）認知有用性

在訪談之初，首針對受訪者對流程調適相關系統或工具的使用經驗來詢問受訪者其公司是否有類似 OntoSPRS 或是否有用過類似的知識管理工具來進行軟體流程調適的經驗。對此，受訪者表示：

「雖然我們有用一些專案管理的軟體工具來輔助，像紀錄專案過程、角色分配、工作分配等等，但在軟體流程調適這方面，我們仍是各憑自己的經驗，靠人工去做調適。」（營運長）

本研究依據首訪者的答案並進一步針對受訪者公司的專案調適現況來做探究。主要目的是為了進一步了解受訪者公司在過去是如何進行專案的調適。針對此問題，受訪者表示：

「就像剛剛說的，我們是靠開發人員的經驗去調適。譬如，今天有一個新的專案進來，我們就會讓比較有經驗的專案經理去決定這個專案應該要怎麼調適。因為有經驗的專案經理，他有經驗，比較知道專案會遇到什麼樣的情境和問題，所以也比較知道流程上可以怎麼調適，譬如專案規模的大小，如果規模不大的話，就可以把一些流程省略掉。專案經理會跟工程師討論，最後來決定調適的結果。」（營運長）

根據受訪者的回答，在沒有工具輔助的情況下，受訪者公司因此只能靠資深人員的經驗與知識來進行軟體流程的調適。而調適的結果也將隨著執行人員的能力與經驗的不同而有不同的展現。由上而知，受訪者公司極度的依賴資深人員的經驗來進行流程的調適。就人員經驗，也就是知識議題進一步追問，一旦這些資深人員離職，將會對公司造成何種影響。受訪者表示：

「這影響一定很大。假設今天換成一位經驗比較不足의專案經理來做調適，他對專案可能遇到的情境不是那麼了解，那他做出來的調適就不會那麼到位。有些程序可能可以被簡化，因為專案經理沒經驗，不知道，那這些程序留下來，就會變成開發人人員作上的負擔，但對專案本身可能沒有太大的幫助。或者說，專案經理不小心把一些必要的程序省略掉，這可能就會造成開發過程的問題，影響到專案進行的順暢度，甚至是整個專案的成效。」（資訊處副理）

目前國內企業的軟體專案人力比例是相對不足的，當人員負荷過重且工作內容又高度依賴經驗與知識時，這將容易導致作業延宕而最終影響到軟體專案的成效並造成公司的損失。在了解受訪者的過往經驗及公司現況後，便進一步讓受訪者針對 OntoSPRS 來發表其看法。其認為：

「這個系統能根據現在準備要調適的專案的狀況，例如專案的規模、專案團隊的大小，系統用過去所累積的調適經驗，來推薦我們應該要怎麼做、怎麼調適。當然，每個專案都有不同的環境條件，過去可能適合這樣調適，但現在不一定適合。這個系統有考慮到這一點，讓使用者自己決定採納或不採納系統推薦的

調適建議。再來，這個系統能把過去的調適結果記錄下來，比較沒經驗的人員就可以有一個調適的參考依據。」(軟體工程師 A)

總歸以上，可以發現受訪者亦認同本研究系統之機制，可用來改善其調適建議。另外，受訪者表示 OntoSPRS 對其工作是有幫助的。最後，針對系統的有用性並讓受訪者做最後的結論。受訪者表示：

「我覺得這個系統最大的優點就是可以利用過往的經驗來推薦出調適建議，就算今天不得已必須讓一個經驗比較不足的專案經理來做調適，利用這個系統，就可以提供專案經理調適建議來做參考，來彌補他經驗上的不足。」(資訊處副理)

(二) 認知易用性

本研究欲了解受訪者對易用性的看法並詢問受訪者，OntoSPRS 的易用性是否會影響其使用本研究系統的意願。受訪者表示：

「這影響很大。系統操作的難易程度會影響我們願不願意接受和使用這個系統。如果系統的使用者介面不夠直觀或很難使用，就會降低我使用這個系統的意願，甚至是排斥它。導入系統本來就是為了提升我們的工作效益，而不是增加我們的工作負擔。」(軟體工程師 A)

從以上的回答，可瞭解到受訪者非常重視系統的易用性。然而，在某些情況下，有些系統確實提供了很強大的功能，但在操作介面上卻非常難以上手；因此，進一步了解受訪者對於易用性低但可助於其完成任務的資訊系統，將會如何影響其使用意願。其表示：

「我們會排斥這樣的一個系統，甚至不想使用它。即使這個系統再好，如果它不容易使用和操作，我也無法感受到它的好處和好用在哪裡。再來，如果它是一個很有用的系統，但卻非常難用，公司可能就得花很大的成本去推廣或宣導這個系統。我們人員也得付出很多時間和努力去學習怎麼使用這個系統。」(資訊處副理)

在瞭解受訪者對易用性的看法後，便進一步讓受訪者針對 OntoSPRS 的易用性來發表其看法。對此，受訪者表示：

「以我使用的感覺，使用者介面的操作算是滿簡單易懂的。而且操作上不會太困難，只要有網路的地方就可以連線上來使用。流程調適的過程，系統也會一步一步的逐項引導我們如何去操作。」(軟體工程師 B)

(三) 主觀規範

由以上訪談內容來看，受訪者對 OntoSPRS 雖有使用意願，但是否會繼續使用於後續專案內，仍未明確表達。另外，OntoSPRS 系統的於個案公司實作過程中，一開始即由高階主管指示使用，故 OntoSPRS 系統的使用情境基本上是強制

的，也就說個案公司人員對系統使用是非自發性的。為瞭解個案公司之主觀規範對 OntoSPRS 使用意圖，再次詢問其看法與繼續使用意願。對此，受訪者表示：

「過去，我曾經指示專案經理去調查有關專案流程調適的工具，以簡化其專案的調適工作。後來他們向我回報，國內外並沒有專門針對專案流程這方面的工具；至於 open source，僅有針對開發的功能點數量相關的預估軟體，對本公司的專案並不合適，而且使用者介面非常地不 friendly…。然而，在本次專案，專案成員（作者按：專案經理與工程師）都對你們的系統感到有興趣，使用者介面非常友善，而且操作相當容易…。所以感覺上比較符合專案成員想要的…。」（營運長）

（四）使用意圖

從以上的訪談內容，可瞭解到個案公司曾受主管人員影響，過去也曾思考過以系統輔助專案成員進行流程調適工作，但對市場上專案管理工具所提供的功能或操作介面，個案公司人員雖曾使用過，並無繼續使用之意願；因此，進一步瞭解實際專案成員對於 OntoSPRS 繼續使用之意願，其表示：

「當我們在參與專案流程活動時，若有輔助系統可以協助我們較容易完成調適工作，我們會非常樂意使用的，並且會願意花費更多的時間瞭解與使用該系統。該系統可較易協助解決問題，讓專案成員認知系統具有使用價值，也就是提高對系統對流程調適工作有用性之認知，進而提高專案成員對系統的後續使用意願…。」（資訊處副理）

綜合認知有用性、認知易用性、主觀規範與使用意圖的訪談內容分析，個案公司成員對於 OntoSPRS 的操作並沒有太大的障礙，也說明了認知易用性確實會正向的影響認知有用性，進而間接影響使用者的使用行為意圖。整體而言，受訪者對於 OntoSPRS 的有用性、易用性與使用意圖皆抱有正向的態度。

二、知識本體架構分析

本研究除 OntoSPRS 建置外，另一重點在於軟體流程知識本體之建構，此節針對所開發的知識本體架構來進行分析與討論。Chi（2010）認為，本體論的建置並無所謂的絕對正確與否，而是端看是否能夠符合其領域的應用而定。雖如此，本研究認為仍可藉由其他要素的衡量來確保建置出的知識本體之結構具擴充性與重複使用之效益。Chi（2010）指出，建置知識本體之一個重要目的即是在所屬的專業領域具有重複利用之價值。由於知識本體的建置是（知識）累積的結果、是一種循序且重複的過程；其藉由不斷地修改與新增舊的本體來成就一個更為完整的領域知識。因此，本研究以研究者觀點，針對關於本體之完整性（Recker et al. 2011; Oh et al. 2011; Tsai et al. 2013）與一致性（Recker et al. 2011;

Tsai et al. 2103; Haghighi et al. 2013) 等評估構面，檢定分析本研究所建構之知識本體。

(一) 完整性 (Completeness)

為使知識能被永續利用，知識本體需要定期維護並持續擴充與修訂，因此「不完整」是本體論根本上的限制，故在定義上並無法確保其內容之完整性 (Recker et al. 2011)。雖是如此，仍可從反面來探討其目標知識本體是否完整，亦即知識本體是否能滿足使用者的知識需求 (Tsai et al. 2013)。另外，Recker et al. (2011) 與 Oh et al. (2011) 指出，知識本體的完整性可以從其理論依據來討論。在本研究的知識本體中，核心架構為調適概念。本研究透過第貳章的文獻探討蒐集本體中的調適概念，並進一步歸納出十五項專案調適因子及五項流程調適作法；並且透過系統實作與個案公司實例，可不斷地記錄與累積專案知識，使本研究建置之軟體流程知識本體趨於完整，並可做為個案公司未來專案調適之依據或參考。因此，當下本研究之知識本體架構其軟體流程調適概念與作法是完整的；唯軟體開發流程方法 (例如：PMBOK 已發行第五版、CMMI 更新至 V1.3) 推陳出新，需不斷完善與擴充，方使本研究之軟體流程調知識本體日益完整。

(二) 一致性 (Consistency)

依據 Recker et al. (2011) 所述，一致性定義為知識本體的設計內容與應用結果是否產生矛盾。本研究依據 Chi (2010) 所提出的本體建置方法來設計其本體模型，其組成包含了領域本體、任務本體、語意規則，茲就這三個構件探討本體設計要點與一致性：

領域本體：本研究針對軟體專案的產品、專案、流程及人員等類別，定義一般性的概念及實例，領域本體以建置通用性的分類架構為原則，利用 has-a 關係建立具有階層性的從屬關係，協助定義軟體流程領域知識的框架。領域本體主要協助與使用者的溝通，特別是提供一致性的術語定義，建立調適用的資料，包括十五項調適因子及五項調適作法。

任務本體：本研究針對研究建立「調適建議」這項概念，就軟體生命週期、流程作業、作業程序、流程產出人員角色等內涵知識，分別建立跨概念的對應關係、從屬、及組成邏輯。表 2 為上述各概念的屬性定義，其中標示 object 的屬性，用於描述概念及概念之間的 has-a 關係，它是任務本體中最重要的一致性定義；另一方面，屬性也可依內容的獲得方式區分為 asserted 或 inferred，其中標示 asserted 之屬性僅有一項，須在實作階段收集之現況事實，標示 "inferred" 之屬性共六項，它們是設計做為解決問題之用，其內容值雖暫時無法獲得，但可藉由其他已知事實得佐證，由推論機制協助獲得。

語意規則：在表 3 推論規則中，針對標示為 inferred 之屬性，取得調適建議

的語意規則，首先利用規則分析模擬取得調適建議之步驟，各步驟須是已知的事實知識，最後再將規則分析以邏輯式統一寫成「 $(atom_1, atom_2, \dots, atom_n) \rightarrow consequence$ 」之格式，也就是邏輯格式一致性，規則須由推論引擎執行推論。

另外，針對本體內容與應用結果的一致性，Haghighi et al. (2013) 與 Tsai et al. (2013) 建議可從使用者使用系統後的感受或使用者經驗來進行探討，故本研究之知識本體模型除定義軟體流程架構及相關調適之概念外，並在系統實作階段由個案公司所提供實例，且輸入的實例皆已通過實證；亦即，其內容是經實際的專案開發並獲客戶所認同（驗收通過），並確認實例內容與本體中概念之間的一致性。

陸、結論與未來展望

對於軟體專案而言，成功的關鍵在於能否真正掌握調整項目，並進一步落實到專案流程上。本研究針對軟體專案流程調適提供了一個知識管理架構，並進一步建置成一個實際使用的系統。在學術上，本研究展現了本體論的另一應用價值—即軟體流程與專案管理上，提供一個流程調適的知識框架以供其他研究者參考。有關建立知識本體的方法在目前研究中尚屬難解之議題；而本研究研究之軟體流程本體則提供了在建立程序上的建議，亦即從流程調適類別展開至可操作的調適建議、然後建立調適作法與調適建議的對應，以利抽象的軟體流程調適可具體地展現在系統上。在實務上，本研究所提出的系統可使專案成員在軟體開發過程中能將伴隨的調適考量加入至設計中，使得目標專案之流程即時掌握與確認，讓專案成員知識即時分享並可調整現有知識規則。知識規則庫經由不斷地儲存與加值應用，能更增添其價值與成熟度，也將人員知識進一步轉化為組織知識。

由於本研究所發展的系統雖然已在個案公司使用，但仍屬於研究雛形，尚有許多研究限制與須進一步改善之處。首先，本研究所建立的系統，其內容主要是針對系統開發專案為例，在未來若運用上，需視組織的特性與專案任務類型來客製本體內容；其次，關於本體實例的建置與永續擴充，本研究所接觸之企業以物流相關系統為主，是使用該企業之過往資料來進行實例建置，對於其他開發領域不見得可直接採用；另外，本體的建置本身即是專業與特定知識收集的過程。為了能建立完整而適切的本體模型，本研究建議在應用時仍需要透過本體專家的協助、並配合組織的知識提供者來共同建立。最後，則是有關系統的永續改善。本研究也將繼續研發所建置之系統，特別是期望在未來於雲端環境中執行，以更能展現系統的便利性與實用性。

參考文獻

- 陳仲儼、黃永福 (2009), 『運用模糊專家系統來協助軟體專案進行流程調適』, 電子商務學報, 第十一卷, 第一期, 頁 235-258。
- Aaen, I. (2008), 'Essence: facilitating software innovation', *European Journal of Information Systems*, Vol. 17, No. 5, pp. 543-553.
- Alegria, J.A.H., Bastarrica, M.C., Quispe, A. and Ochoa, S.F. (2014), 'MDE-based process tailoring strategy', *Journal of Software: Evolution and Process*, Vol. 26, No. 4, pp 386-403.
- Chen, C.Y. and Tsai, W.L. (2011), 'Toward Single-sourcing of Software Project Documented Contents: A Preliminary Study', *International Journal of Electronic Business Management*, Vol. 9, No. 1, pp. 37-45.
- Chen, C.Y., Liu, H.A. and Song, J.Y. (2013), 'Integrated projects planning in IS departments: a multi-period multi-project selection and assignment approach with a computerized implementation', *European Journal of Operational Research*, Vol. 229, No. 3, pp. 683-694.
- Chi, Y.L. (2010), 'Rule-based ontological knowledge base for monitoring partners across supply networks', *Expert Systems with Applications*, Vol. 37, No. 2, pp. 1400-1407.
- Chi, Y.L. and Chen, C.Y. (2009), 'Project teaming: Knowledge-intensive design for composing team members', *Expert Systems with Applications*, Vol. 36, No. 5, pp. 9479-9487.
- Clarke, P. and O'Connor, R.V. (2012), 'The situational factors that affect the software development process: Towards a comprehensive reference framework', *Information and Software Technology*, Vol. 54, No. 5, pp. 433-447.
- CMMI Product Team (2010), *CMMI® for Development Version 1.3- Improving processes for developing better products and services*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA.
- Darke, P., Shanks, G. and Broadbent, M. (1998), 'Successfully completing case study research: combining rigour, relevance and pragmatism', *Information systems journal*, Vol. 8, No. 4, pp. 273-289.
- Ebert, C. and De Man, J. (2008), 'Effectively utilizing project, product and process knowledge', *Information and Software Technology*, Vol. 50, No. 6, pp. 579-594.
- Elbanna, A. (2013), 'Top management support in multiple-project environments: an in-practice view', *European Journal of Information Systems*, Vol. 22, No. 3, pp. 278-

294.

- Ginsberg, M.P. and Quinn, L.H. (1995), *Process Tailoring and the Software Capability Maturity Model*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, USA.
- Haghighi, P.D., Burstein, F., Zaslavsky, A. and Arbon, P. (2013), 'Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings', *Decision Support Systems*, Vol. 54, No. 2, pp. 1192-1204.
- Horrocks, I., Patel-Schneider, P.F., Bechhofer, S. and Tsarkov, D. (2005), 'OWL rules: A proposal and prototype implementation', *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 3, No. 1, pp. 23-40.
- Kaur, R. and Sengupta, J. (2013), 'Software process models and analysis on failure of software development projects', *International Journal of Scientific and Engineering Research*, Vol. 2, No. 2, pp.1-4.
- Kalus, G. and Kuhrmann, M. (2013), 'Criteria for software process tailoring: a systematic review', *Proceedings of the 2013 International Conference on Software and System Process (ICSSP 2013)*, San Francisco, USA, May 18-19, pp. 171-180.
- Liu, J.Y.C., Chen, H.G., Chen, C.C. and Sheu, T.S. (2011), 'Relationships among interpersonal conflict, requirements uncertainty, and software project performance', *International Journal of Project Management*, Vol. 29, No. 5, pp. 547-556.
- Lorenz, W.G., Brasil, M.B., Fontoura, L.M. and Pereira, G.V. (2014), 'Activity-Based Software Process Lines Tailoring', *International Journal of Software Engineering and Knowledge Engineering*, Vol. 24, No. 9, pp. 1357-1381.
- Myers, M.D. and Newman, M. (2007), 'The qualitative interview in IS research: Examining the craft', *Information and Organization*, Vol. 17, No. 1, pp.2-26.
- Na, K.S., Simpson, J.T., Li, X., Singh, T. and Kim, K.Y. (2007), 'Software development risk and project performance measurement: Evidence in Korea', *Journal of Systems and Software*, Vol. 80, No. 4, pp. 596-605.
- Nidumolu, S.R. (1996), 'Standardization, requirements uncertainty and software project performance', *Information & Management*, Vol. 31, No. 3, pp.135-150.
- Oh, S., Yeom H.Y. and Ahn, J. (2011), 'Cohesion and coupling metrics for ontology modules', *Information Technology and Management*, Vol. 12, No. 2, pp. 81-96.
- Park, S., Na, H., Park, S. and Sugumaran, V. (2006), 'A semi-automated filtering technique for software process tailoring using neural network', *Expert Systems with Applications*, Vol. 30, No. 2, pp. 179-189.

- Park, S.H. and Bae, D.H. (2013), 'Tailoring a large-sized software process using process slicing and case-based reasoning technique', *IET Software*, Vol. 7, No. 1, pp. 47-55.
- Prikladnicki, R., Audy, J.L.N., Damian, D. and de Oliveira, T.C. (2007), 'Distributed software development: practices and challenges in different business strategies of offshoring and onshoring', *Proceedings of the 2nd IEEE International Conference on Global Software Engineering (ICGSE 2007)*, Munich, Germany, August 27-30, pp. 262-274.
- Recker, J., Rosemann, M., Green, P.F. and Indulska, M. (2011), 'Do Ontological Deficiencies in Modeling Grammars Matter?', *MIS Quarterly*, Vol. 35, No. 1, pp. 57-79.
- Reich, B.H., Gemino, A. and Sauer, C. (2012), 'Knowledge management and project-based knowledge in it projects: A model and preliminary empirical results', *International Journal of Project Management*, Vol. 30, No. 6, pp. 663-674.
- Samuel, B.M., Watkins, L., Ehle, A. and Khatri, V. (2015), 'Customizing the Representation Capabilities of Process Models: Understanding the Effects of Perceived Modeling Impediments', *IEEE Transactions on Software Engineering*, Vol. 41, No. 1, pp. 19-39.
- Seaman, C.B. (1999), 'Qualitative methods in empirical studies of software engineering', *IEEE Transactions on Software Engineering*, Vol. 25, No. 4, pp. 557-572.
- Simperl, E., Bürger, T., Hangl, S., Worgl, S. and Popov, I.O. (2012), 'ONTOCOM: A reliable cost estimation method for ontology development projects', *Journal of Web Semantics*, Vol. 16, pp. 1-16.
- Singh, R. (1996), 'International Standard ISO/IEC 12207 Software Life Cycle Process', available at <http://www.abelia.com/docs/12207cpt.pdf> (accessed 15 April 2015).
- Sulayman, M., Mendes, E., Urquhart, C., Riaz, M. and Tempero, E. (2014), 'Towards a theoretical framework of SPI success factors for small and medium web companies', *Information and Software Technology*, Vol. 56, No. 7, pp. 807-820.
- Tsai, W.T., Wu, B., Jin, Z. Huang, Y. and Li, W. (2013), 'Ontology patterns for service-oriented software development', *Software: Practice and Experience*, Vol. 43, No. 7, pp. 867-883.
- Martínez-Ruiz, T., Münch, J., García, F. and Piattini, M. (2012), 'Requirements and constructors for tailoring software processes: a systematic literature review', *Software Quality Journal*, Vol. 20, No. 1, pp. 229-260.
- Venkatesh, V. and Davis, F.D. (2000), 'A theoretical extension of the technology

- acceptance model: four longitudinal field studies', *Management Science*, Vol. 46, No. 2, pp. 186-204.
- Williams, L. and Cockburn, A. (2003), 'Agile software development: It's about feedback and change', *IEEE Computer Society*, Vol. 36, No. 6, pp. 39-43.
- Xu, P. and Ramesh, B. (2007), 'Software process tailoring: an empirical investigation', *Journal of Management Information Systems*, Vol. 24, No. 2, pp. 293-328.
- Xu, P. and Ramesh, B. (2008), 'Impact of Knowledge Support on the Performance of Software Process Tailoring', *Journal of Management Information Systems*, Vol. 25, No. 3, pp. 277-314.
- Yan, T. and Dooley, K.J. (2013), 'Communication intensity, goal congruence, and uncertainty in buyer-supplier new product development', *Journal of Operations Management*, Vol. 31, No. 7, pp. 523-542.
- Yan, T. and Dooley, K.J. (2014), 'Buyer-Supplier Collaboration Quality in New Product Development Projects', *Journal of Supply Chain Management*, Vol. 50, No. 2, pp. 59-83.
- Young, R. and Poon, S. (2013), 'Top management support-almost always necessary and sometimes sufficient for success: Findings from a fuzzy set analysis', *International Journal of Project Management*, Vol. 31, No. 7, pp. 943-957.

