一個基於知識本體論之智慧型專案管理系統

An Ontology-based Intelligence Project Management System

鄭為中, 黃科瑋, 蔡崇煒, *楊竹星
國立成功大學電機工程學系暨電腦與通訊工程研究所
*csyang@ee.ncku.edu.tw

***摘要*— 知識本體論或稱本體論(Ontology)，其主要目的為描述特定的知識領域。本體論的優勢在於能夠提供一個更簡易的方式，讓人與機器溝通，以及系統與系統之間交換資訊。透過本體論，我們設計了一個智慧型專案管理系統 (Intelligent Project Management System; IPMS)，除利用本體論對系統作詳細的描述，並且使用本體論語言-OWL(Web Ontology language) 與其提供的推論引擎，定義出相對應的規則使管理系統可以針對不同的情境作出自動化的推理。其中包含進度的控管及會議時間的選擇。此外，IPMS 可自動處理在規則定義底下的事件，進而達成一個具有智慧的智慧型專案管理系統。**

***關鍵詞*— 本體論，本體論語言，情境，推論引擎。**

***Abstract*—In general, the ontology is used to describe domain knowledge and to share it. The advantage of ontology is to provide a simple way between human and computers connection, and makes it easier to exchange the information of different systems. Based on the ontology, we present an Intelligent Project Management System (IPMS) which uses the OWL (Web Ontology language). This OWL defines the corresponding rule and makes the management system solve the different situations which include the controlling of the time automatically. In addition, IPMS can process the following incident in the rule, and accomplish intelligent project management systems.**

***Keywords* — Ontology, OWL, Context, Reasoning.**

一、緒論

近年來，由於網路世界已經成為人類生活不可缺少的一部分，所以對於資訊的儲存及管理，都與網路有分不開關係。由 Berners-Lee 等人提出的語意網和其架構[13,16]就是針對網路資訊倍增的一個處理機制，使電腦具有智慧去解讀網路上的資訊，並加以自動化的處理。知識本體論[14] (Ontology)是構成語意網最重要的部份，藉由使用類別和關係的定義，使得大部分的資料與行為及情境能被定義之。透過本體論，不僅可讓電腦和人可以有更簡易的溝通路徑，系統與系統之間也能有更方便的傳遞訊息的方式。

隨著本體論語言中的 OWL[6-7] (Web Ontology Language)不斷的增修與進步，舉凡像是使用知識本體來挑選適合的分類[18]或是結合了本體論和資料探勘的知識呈現系統 ParkaDB[17]。都搭配使用了本體論來呈現其專案領域(domain ontology)的概念(Concepts)，以及情境(context)和詞彙(vocabulary)的關係。傳統上的專案管理系統，依賴人與電腦的溝通。當專案管理包含了跨公司或是多個部門，專案管理者控管專案的難度相對的就提高許多。針對這

個問題，在本篇論文中我們提出一個基於本體論的智慧型專案管理系統 IPMS[5] (intelligent Project Management System)。使用本體論的觀念來定義整個系統相關的概念以及關係，並使用 OWL 來達成分享情境的能力。IPMS 結合了推論的功能，使得原先須經由人工去處理的問題，像是進度的控管和會議開會時間的選擇，推論軟體經由當下的情境與所定義之規則，推論出所相對應的決策。增加了具推論功能的 IPMS，也讓一般 OWL 擴充至擁有情境推論的能力。

本論文的章節架構說明如下:首先在第二章介紹相關的研究文獻;第三章定義出 IPMS 的整體系統架構;接著在第四章詳細描述系統中的主要元件與推論的部份;最後是結論與未來的研究方向與工作。

二、相關研究探討

(一) 知識本體論(Ontology)

知識本體論(Ontology)此一術語原是在哲學領域上探討宇宙萬物及探究其本質的一門學問。而在近幾年來由W3C組織將本體論這個字彙定義為用來描述各種專業領域的描述，並大量的應用在語意網(Semantic Web)及知識管理(knowledge management)等研究上。透過明確定義特定領域中的概念(concept)和概念間的關係(relation)，可清楚的表達出對所要定義的領域(domain)其語意(context)，詞彙(vocabulary)，和資訊(information)。為了使電腦和設計者所建構出的知識本體溝通，需要讓電腦清楚明白所建構的不同領域間不同的知識本體。所以W3C開發出一套能使電腦理解之語言用來轉換描述知識本體論語言，也就是OWL(Web Ontology Language)。

(二) 本體論語言

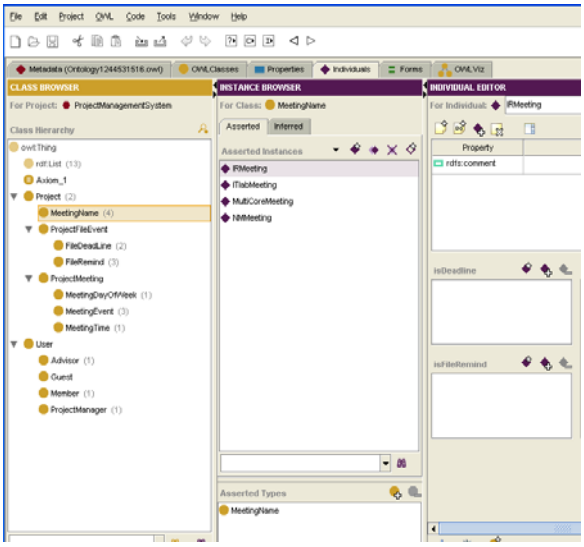
為了要建立一個完整的本體論，我們需要讓電腦透過本體論語言去了解所建構的本體論。首先 Uschold[19]定義了 Ontology 是由實體(entities)、屬性(attributes)以及關聯(relations)三個觀念所組成。 此外 Ontology 擁有 Knowledge Sharing、Logic Inference 以及 Knowledge Reuse 等特色，所以採用來建構語意網(Semantic Web)是十分合適的。隨後 W3C 也根據語意網，提出了 RDF[9-10] (Resource Description Framework) ， RDF Schema[11] ， XML[1](extensible Markup Language)以及 XML Schema[15]來做為開發語意網的語言。一般認知上的 OWL(Web Ontology Language)也就是一種 XML-based 的文件，W3C 根據 OWL 也提供了三種表達能力的子語言，包含 OWL-Lite、OWL-DL 和 OWL-Full。針對 W3C 組織提出的 XML、XML Schema 、RDF 以及 RDF Schema 的描述如下所示:

1. XML and XML Schema : XML 是提供結構化文件的表現語法。而 XML Schema 是針對每一個 XML 文件結構加以限制的語言。

2. RDF and RDF Schema : RDF 是定義資源和資源彼此之間關係的資料模型，而 RDF 文件也是使用 XML 格式來定義。RDF Schema 則是定義了 RDF 在特定領域中的屬性(property)和類別(class)的字彙。

(三) Protégé

Protégé[8]是由 Stanford Medical Informatics Group 所開發，提供了 Protégé-OWL editor 以及 Protégé-OWL API 的使用者界面。這套軟體的建立目的除了簡化使用者對於本體論開發流程外，也提供了直觀的圖形化方式以供使用者觀看。選擇使用 Protégé 來進行 Ontology 的開發，原因是如果未來需要進行與其他現有的知識本體系統做結合的動作，使用 Protégé 來整合不同系統可大量減少系統整合花費的時間與成本。



圖一.Protégé 操作界面

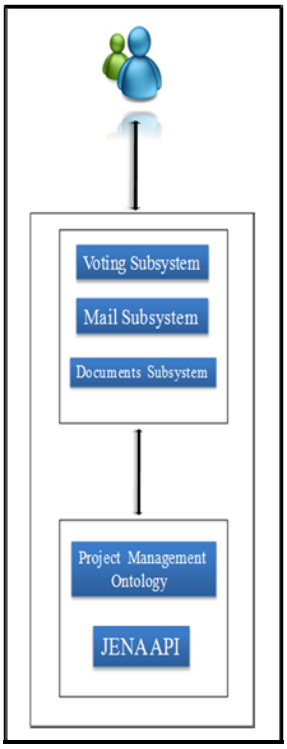
(四) Jena API

Jena API[4] 是由HP公司所開發的一個支援Java的程式語言開放性資源，一般稱為Jena API。目的是為了針對RDF、RDF-Schema OWL-Lite、OWL-DL和OWL-Full 等幾種不同本體論語言提供一個規則為基礎的推論引擎。而除了推論的功能之外，還可以進行RDF文件查詢功能，而Jena使用了SPARQL[12]來做為RDF query 使用的語言。針對推論而得的結果，使用XML Parser將可更精確的獲得所需的內容。故透過XML Parser以及Jena所提供的推論功能，可讓系統更具智慧性，以達到智慧型管理系統的目標。

三、系統架構

(一) 系統環境

我們使用的平台是 Windows XP SP2, 使用的 Ontology 建設環境是 Protégé API, MySQL 資料庫被用來存放規則和系統所有的資料。在使用者界面設計上，IPMS 使用了 JSP 這個語言。後端推論引擎則是 Jena API。



圖二. IPMS 系統架構圖

(二) 系統整體架構

目前的專案管理系統大都以人工為主，對於複雜的專案並無非常可靠的一套管理模式。如果將時間、地點以及情境的資訊加入專案管理系統，再透過推論引擎，是可讓系統自動化的去管理計畫。根據上述的動機，我們的構想是定義一個結合知識本體論關念的智慧型專案管理系統，運用本體論的觀念，設計出一具有智慧的自動化專案管理。

為此首先透過 Protégé 來定義出一個 Project Management Ontology，透過三個前端子系統搜集來的情境最後由推論引擎 Jena 進行推論出一個事件(event)再由系統自動執行之，目的是透過 Web Service 自動化的管理方式。圖二呈現 IPMS 系統的整體架構，分為二個部份，前端處理模組。與包含三個子系統(Mail、Voting、Documents subsystem)。Voting 子系統是在處理針對專案會議時間的投票。根據會議主持人的行程經由推論引擎推薦給會議主持人一個對於眾人皆較恰當的時間，最後由會議主持人確認時間；Mail 子系統是主要處理針對郵件處理的部份，Documents 子系統提供專案文件管理系統。另一

個部份就是後端推論的模組，主要包括推論引擎設計的 Project Management Ontology，主要是用來定義出 IPMS 所需要使用到的各種類別與其關係。Jena API 則針對前端子系統的情境推論出一個事件讓系統執行之，以求達到自動化管理的目的。

四、系統實作與設計

在 IPMS 系統的設計上，首先基於 OWL 語言，利用 Protégé 建構出 OWL，透過 Jena 進行推論，並在後端更新資料庫，最後前端經由使用者界面來呈現結果。在設計初期，我們選定使用 Jena 作為我們後端的推論引擎，為了與 Jena 做更好的連結我們使用 JSP 作為我們前端 User Interface 部分的製作，而在 Jena 使用上的問題我們也多參照由 Jena 開發團隊成立於 Yahoo Group 的一個 Jena Tech Group[20]。另外 Jena 推論所得之結果是以 RDF 格式呈現，針對此格式，我們也透過 RDF parser，擷取出我們所需要的資訊，方能更順利的完成我們的推論及其後續的動作。

(一) Project Management Ontology

首先系統將使用者一身份分成四個層級，分別為 Guest、Member、Project Manager 以及 Administrator。Guest 代表的是一般訪客，完全不具有任何權限，不能存取任何一個相關的專案資料，僅能存取系統提供的公開資料。Member 代表的是各別計畫底下的成員，此階層只能存取所屬於自己專案的資料，也只能下載自己組員的資料及文件，但並無修改的權限; Project Manager，顧名思義，就是專案的管理員，擁有該專案所有管理的權限。Administrator 可以控管所有的計畫內容，也就是具有最高權限的使用者。系統將 Project 以階層的方式表示，分成 Project_Name、Project_State 及 Project_Meeting 三種階層，如以下所述:

(1)ProjectName 就是表示計畫的名稱。
(2)ProjectState 針對 Porject 的描述：計畫支援機關或單位(如果科會、教育部等)、計畫期限(開始日期、截止日期)、計畫經費、參與人員分工情形、計畫摘要以及計畫 KPI(預期達成目標)等，也包含 FileUploadDate 以及 FileUploadMember 而 FileUploadDate 還有一個子類別為 UploadDeadLine 是用來判斷是否已經超過上傳期限，主要是用來判斷成員是否有準時上傳檔案，並且檔案所有者名稱。
(3)Project_Meeting 也擁有 MeetingDayOfWeek 這個子函式。用來決定要選取什麼日期時候 Meeting，MeetingTime 此函式決定 Meeting 的詳細時間，而 MeetingEvent 是用來解決對計畫成員或計畫管理員進行 Meeting 時突發情況的應對措施。

此外，此系統定義了一些關係來針對整個 Project Management System 做更詳盡的描述如 (weekly report 的機制) 此機置是為了讓管理者可以直接透過 weekly report 和 user 互動。

圖三.系統部分 OWL 內容

(二) 會議系統子系統

IPMS 設計了三個子系統，第一個是 Mail system，此子系統的目的就是為了要通知計畫成員所有資訊，如提醒功能(提醒使用者需上傳該次報告文件或是週報文件)和通告功能(告之使

用者什麼計畫於何時何地舉行會議)。第二個子系統是投票機制的子系統，由於會議可能會臨時有更動時間，如果需要更動時間，則系統會自動由 Mail system 發出信件通知會議參與成員至 Voting system 選取可能的投票時間，最後再由系統自動決定時間通知各成員。最後一個也就是檔案管理子系統，負責管理所有文件檔案並由系統自動更新會議報告名單。

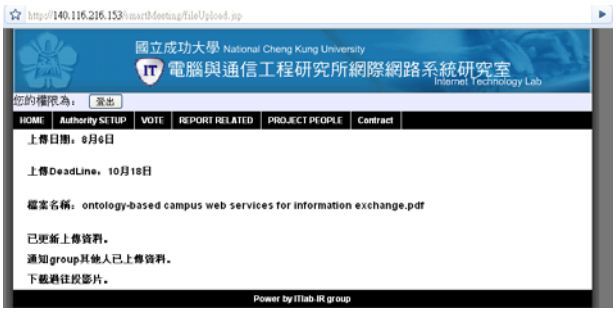
(三) 推論引擎

設計推論引擎的目標，就是要讓 IPMS 系統達到自動化管理。後端推論引擎讓三個子系統達到情境感知的功能。以下為 IPMS 可達到的四個情境感知實例。

(1)確定會議時間:首先由會議主持人先選定幾個合適的會議時間後通知系統後會由 Jena 通知 Mail system 寄件給專案參與成員。經由大家的投票後由系統推論出一些較合適的時間，此時間亦會經由推論系統根據會議管理者的行程，排除行程上較不可行的會議時間，最後推薦管理者時間較可行的會議時間，交由管理者決定。再回覆給每一個專案成員，如圖四所示。

圖四.經由系統推論並推薦會議時間

(2)通知檔案上傳: 假設會議報告者未上傳相關文件至系統，當設定時限到達，文件管理系統由 Jena 推論出未上傳之事件已產生，而 Mail system 發送通知給報告者。文件上傳後，檔案系統會由 Jean 和 Mail system 通知該自動發信給其所屬的專案成員，如圖五所示。



圖五. 文件上傳後自動通報

(3)自動更新會議報告人:系統會根據時間判斷，然後由 Jena 通知 doc system 系統更新下次會議報告成員和發信給下次會議需報告的人和所屬成員。

(4) Weekly Report，主要是針對專案成員的進度掌控所設計，由於每位專案成員不僅需要在會議時報告，會議後需訂製文件與完成進度報告檔案上傳。然而在管理者閱讀完進度報告後可馬上透過 doc system 填寫建議事項再由 Mail system 通知該成員觀看與處理。藉由其 Weekly report 我們期待可以達到良好的 user 和其管理者的互動與進度管控，如圖六所示。

圖六.Weekly Report 格式及自動提醒功能

最後，我們用一簡單範例來說明推論的規則，如表一所示。在 FileRemindYes 以及 FileRemindNo 規則中，當 x 同學屬於 Meeting y 而且為排定簡報人選，若 x 同學在期限內尚未

上傳檔案至 Meeting y 的資料夾中，即可推論出 Meetiny y 的上傳提醒狀態為 Yes；反之，其上傳提醒狀態為 No；在 MeetingState 處理中，負責管理 Meeting(Project)的 Manager 若當天的出席狀態為無法出席(AttendNo)或是延遲出席(AttendDelay)則對應將 MeetingState 設成 MeetingStateCancel 或 MeetingStateDelay 並且由系統發信通知參與 Meeting 的 Member。

表一.簡單推論規則(Rule)範例

FileRemindYes
(?x itlab:Join ?y), (?x itlab:isFileUploadNo ?y) -> (?y itlab:isFileRemind itlab:FileRemindYes)
FileRemindNo
(?x itlab:Join ?y), (?x itlab:isFileUploadYes ?y) -> (?y itlab:isFileRemind itlab:FileRemindNo)
MeetingStateCancel
(?x itlab:Manage ?y), (?x itlab:MeetingAttend itlab:AttendNo) -> (?y itlab:MeetingState itlab:MeetingStateCancel)
MeetingStateDelay
(?x itlab:Manage ?y), (?x itlab:MeetingAttend itlab:AttendDelay) -> (?y itlab:MeetingState itlab:MeetingStateDelay)

五、結論與未來研究方向

在本論文中，我們提出一個使用本體論概念並結合了語義網特性的智慧型專案管理系統 IPMS。其使用 Ontology 的觀念來定義整個系統相關的概念以及關係，並以 OWL 語言來達成分享情境的能力。使用 Jena 這套 API 來達成推論的功能，進而使系統可以自動化感知情境並針對專案系統的不同元件去判斷該使用的服務。

日前我們已經完成了多種專案管理系統的應用。未來的研究方向會朝以下二種應用來做改良及擴充，並會加入 User Guide 的部份方便使用者操作系統。

1. 將 IPMS 系統和 Google Calendar[2]做結合，使得系統可以針對可能臨時發生的事件做會議開會時間的調整。預計使用 Google Calendar 所提供的 API，透過蒐集必須的資訊，使 IPMS 可以針對 Manager 的行程，做出時間上的規劃，並協助判斷行程與時間配合的合理性，例如，Manager 中午十二點於台北開會，但會議於下午一點開始，若能透過系統自動找出時間上的盲點，進而通知會議延遲(甚至取消)，即可省去許多人工通知的過程，也希望能夠提供讓使用者能夠有個更方便的介面存取 Google Calendar。

2. 整合 IPMS 和 Google Calendar，預期使用 JADE (Java Agent Development Framework)[3]來做為溝通 IPMS 和 Google Calendar 的 Middleware，且由於 JADE 具有代理人的作用，未來在擴充其他應用，應可使用 JADE 來當成第三方的溝通工具。

六、誌謝

本研究承蒙行政院國家科學委員會計畫 (NSC96-2218-E-006-287 及 NSC 97-2218-E-006-012) 經費部份補助，特此感謝。

參考文獻

[1] Extensible Markup Language(XML), http://www.w3.org/XML/.

[2] Google Calendar, http://www.google.com/calendar.

[3] JADE - Java Agent Development Framework, http://jade.tilab.com/.

[4] Jena – A Semantic Web Framework for Java, http://jena.sourceforge.net/.

[5] IPMS(online), http://140.116.216.153/smartMeeting

[6] OWL Web Ontology Language Overvie, http://www.w3.org/TR/owl-features/.

[7] OWL Web Ontology Language Use Cases and Requirements, http://www.w3.org/TR/webont-req/.

[8] Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/.

[9] Resource Description Framework (RDF), http://www.w3.org/RDF/ .

[10] RDF/XML Syntax Specification, http://www.w3.org/TR/rdf-syntax-grammar/.

[11] RDF Schema, http://www.w3.org/TR/rdf-schema/.

[12] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/.

[13] Semantic Web, http://www.w3.org/2001/sw/.

[14] What is an Ontology? http://www-ksl.stanford.edu.kst/what-is-an-ontology.html.

[15] XML Schema, http://www.w3.org/XML/Schema.

[16] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific Am., pp. 34–43, May 2001.

[17] A. Bernstein, F. Provost, and S. Hill,“Intelligent assistance for the data mining process: an ontology-based approach,” CeDER Working Paper IS-02-02, Center for Digital Economy Research, Stern School of Business, New York University, 2002. Aug.

[18] M. Taylor, K. Stoffel, J. Hendler, “Ontology-based induction of high level classification rules,” Proc. of SIGMOD Data Mining and Knowledge Discovery Workshop, 1997.

[19] M. Uschold“An ontology research pipeline,”Applied Ontology, Vol. 1, No. 1, pp, 13-16, 2005.

[20] Yahoo Tech Group/Jena-Dev http://tech.groups.yahoo.com/group/jena-dev/