

國立交通大學

資訊管理研究所

碩士論文



IT服務管理：運用情境認知之知識支援於事件管理
IT Service Management: Context-aware Knowledge
Support for Incident Management

研究生：周柏村

指導教授：劉敦仁博士

中華民國九十四年六月

IT服務管理：運用情境認知之 知識支援於事件管理

研究生：周柏村

指導教授：劉敦仁 教授

國立交通大學資訊管理研究所

摘要

IT 服務管理(Information Technology Service Management, ITSM)，主要是將 IT 視為一種服務，運用服務管理的概念管理企業內的 IT，藉由提供高品質的 IT 服務，提昇企業的競爭優勢。ITIL 提供了一套 IT 服務管理的實務典範，以流程化的方式，進行 IT 服務管理。其中，事件管理流程負責處理日常營運所發生的突發事件，目的在於迅速地將事件解決或者提供暫時性解決方案(Workaround)，以降低對業務運作的負面影響。ITIL 所定義的事件管理流程中，只針對流程與步驟，以及應該注意的方針及要點，提出實務指引，並未探討如何透過適當的知識支援，以協助事件管理，增進事件處理效率。此外，事件通常有不同的情境，情境資訊的取得，可以協助處理人員掌握事件資訊，事件的情境也會影響事件的處理與解決方法。本研究提出情境認知的事件管理知識支援架構，運用資料探勘技術從事件記錄中發掘出與情境相關的事件管理知識，透過規則推論的方式提供情境認知的事件知識支援，有效掌握事件相關資訊，以協助支援事件管理，增進事件處理效率。

關鍵字：資訊科技服務管理(ITSM)、ITIL、資料探勘、情境認知、事件管理

IT Service Management : Context-aware Knowledge Support for Incident Management

Student: Po-Tsun Chou

Advisor: Dr. Duen-Ren Liu

Institute of Information Management
National Chiao Tung University

Abstract

Information Technology Service Management (ITSM) regards IT as a service to manage IT from service-oriented perspective, whereas enterprises can raise their competence through high quality IT Services. ITIL has defined several processes of managing IT services to provide the guidelines for the best practices of ITSM. The Incident Management process handles incidents arisen from regular IT operations to reduce their negative effects on enterprise operations by providing first-aid resolutions or workaround solutions. ITIL-defined Incident Management process focuses on process activities, procedures, and guiding principles to provide practical instructions for handling incidents. Effective knowledge support for Incident Management has not been investigated. Furthermore, identifying the contextual information of incidents can provide more effective incident management by resolving the incidents according to their contextual information. This work proposes a framework of context-aware knowledge support for Incident Management. Association rule mining is employed to discover contextual knowledge of incidents from historical incident logs. Moreover, rule inferences are employed to infer possible contextual knowledge and resolutions. Consequently, effective knowledge supports for incident management can be achieved by providing needed contextual knowledge and possible resolutions through the rule mining and inferences.

Keyword: ITSM, ITIL, Data Mining, Context-aware, Incident Management

誌謝

時候到了，該是邁向下個階段的時候了！！

兩年的研究所生活匆匆畫下句點，一路走來，感謝許多曾經幫助我的人。首先感謝我的指導教授劉敦仁老師，您像個寶貴的智庫，當我研究遇到瓶頸時，總能給我適當的建議與指導，讓我順利的完成碩士學位，在您嚴謹的研究態度以及豐富學養的耳濡目染下，讓我學習到做研究的能力，成為我一生寶貴的資產。同時感謝口試委員陳恭教授與羅濟群教授，感謝你們於論文口試期間提供許多寶貴的意見，讓本論文更加完善。

感謝實驗室的夥伴，帶給我溫暖的 Lab 生活。美玉、阿土、雅月、怡瑾、孟蓉、錦慧、志偉和純和等學長姐們，謝謝你們在生活與學術上的提攜與指導，尤其感謝亦師亦友的阿土學長，感謝你於論文的不吝分享與指導，生活上的照應以及人生經驗的分享，讓我受益匪淺。感謝同門的師兄弟，秋雯、枕瑋、明立、韋孝，在漫漫的論文寫作過程中，有你們的協助以及一同熬夜、打氣與討論，為苦澀的日子注入不少甘甜，其中，與枕瑋組成的 ITIL 二人組，趕專案的日子，更是令人難忘。此外，感謝北晨、皇志、宛蓉、詩瑩與栩嘉等學弟妹，你們的加入，帶給實驗室活力與歡樂，豐富了我們的生活。感謝羽球所的各位陪我鍛鍊身體，力行「書唸的好，身體也要練好」的不成文宗旨。

感謝所上的地下總司令淑惠大姐，以及專班助理欣欣小姐，你們提供有效率的行政作業，讓我在學校生活無往不利，其中淑惠大姐的在地通服務，更是無懈可擊！！

感謝震震科技對我的栽培與照顧，讓我有機會接觸 ITIL 的這個新領域，感謝 Nick、張博士、Nathan、阿雄與泓文等夥伴，除了佩服你們的專業能力外，你們的傾囊相授，讓我學習到許多課堂上學不到的寶貴知識，實務操作的機會，更讓我增加了不少經驗值。

感謝我最愛的家人，你們是我心中最大的支柱，家的溫暖是我努力的動力來源，這個成就是屬於你們的！尤其特別感謝我的母親，您的諄諄教誨，時時刻刻警惕著我，您無微的呵護，溫暖著我心。最後感謝美君陪我走過最難熬的階段，妳的包容、體貼以及全力的支持，讓我無後顧之憂，沒有妳的陪伴，我無法順利渡過這兩年。

感謝交大土地公爺爺，保佑我在新竹平安順利，感謝交大帶給我美好的回憶，感謝資管所...

時候到了，該起身走人了！人生旅途上，我將永遠記得你們帶給我的一切..

柏村 2005海棠隔夜筆
新竹小窩

目錄

一、緒論	1
1.1 研究背景及動機	1
1.2 研究目的	3
1.3 論文架構	4
二、文獻探討	5
2.1 ITIL	5
2.1.1 服務管理	5
2.1.2 事件管理	6
2.2 情境與情境認知	7
2.2.1 情境	7
2.2.2 情境分類	8
2.2.3 情境認知	9
2.3 資料探勘	9
2.3.1 關聯規則探勘	9
2.3.2 以限制為基礎的關聯規則探勘	10
2.4 本體論	11
2.5 OWL	11
三、情境認知事件知識支援架構	13
四、情境認知事件知識支援模組介紹	15
4.1 情境模組	15
4.1.1 情境模式化	15
4.1.2 事件情境資訊池	21
4.2 資料探勘模組	22
4.2.1 事件報告內容	23
4.2.2 有限制的關聯式規則探勘	24
4.3 推論模組	29
4.3.1 規則式推論	29
4.3.2 事實值	30
4.3.3 規則庫	31
4.3.4 推論引擎與符合的規則	32
4.4 情境認知事件管理模組	35
4.4.1 情境認知事件分類知識支援	36
4.4.2 情境認知事件解決方案知識支援	37
4.4.3 情境認知事件相依關係知識支援	38
五、雛型系統建置與實作	39
5.1 開發工具及平台	39

5.2	情境模組的實作內容	40
5.3	資料探勘模組的實作內容	46
5.3	推論模組與情境認知的事件知識支援模組的實作說明	48
六、	結論與未來研究方向	54
6.1	結論	54
6.2	未來研究方向	55
	參考文獻	56



表目錄

表 4.1 情境特徵例子	17
表 4.2 情境相依關係例子	20
表 4.3 事件情境關係池範例	21
表 4.4 事件報告內容	23
表 4.5 事件報告資料分類	24
表 4.6 知識樣板範例	28
表 4.7 事實值來源	31
表 4.8 規則庫來源	31
表 4.9 關聯規則中CF表示法	32
表 4.10 推論過程中，CF值算法	33
表 4.11 符合的規則後項，以CF值排序	34
表 5.1 系統開發工具與平台	39



圖目錄

圖 3.1 架構全覽	13
圖 4.1 情境模組架構圖	15
圖 4.2 情境特徵本體 (Context Feature Ontologies)	16
圖 4.3 情境相依關係(Context Dependency).....	18
圖 4.4 資料探勘模組架構(Data Mining Module).....	22
圖 4.5 限制式規則形式(Constraint Association Rule Type).....	25
圖 4.6 推論模組架構(Inference Module).....	29
圖 4.7 規則式推論模型(Rule-based Reasoning).....	30
圖 4.8 計算CF值例子	33
圖 4.9 情境認知知識支援模組架構(Context-aware Knowledge Support Module)	35
圖 4.10 情境認知事件分類知識支援例子	36
圖 4.11 情境認知事件解決方案知識支援例子	37
圖 4.12 情境認知事件相依知識支援例子	38
圖 5.1 設計情境特徵類別	41
圖 5.2 設計情境特徵類別Person的資料屬性	41
圖 5.3 設計情境特徵類別SysComponent的子類別	42
圖 5.4 設計情境相依關係UseService	43
圖 5.5 設計類別個體“Mary”與其他情境資訊	43
圖 5.6 以OWL展示子類別關係，以SysComponent為例.....	44
圖 5.7 以OWL展示物件屬性，以ULocatedIn為例	44
圖 5.8 以OWL展示資料屬性，以Role為例	44
圖 5.9 以OWL展示類別個體，以Mary為例	45
圖 5.10 事件報告的資料關聯圖	46
圖 5.11 Weka讀入事件報告的表示畫面	47
圖 5.12 情境相關事件分類的關聯規則分析結果	47
圖 5.13 系統登入畫面	48
圖 5.14 使用者主畫面，顯示使用者情境資訊	49
圖 5.15 使用者“sprite”的情境特徵以及情境相依關係.....	49
圖 5.16 使用者填寫事件申請表格	50
圖 5.17 顯示事件的詳細資料	51
圖 5.18 服務台人員的主畫面	51
圖 5.19 情境認知的事件分類推薦	52
圖 5.20 情境認知的事件解決方案推薦	53
圖 5.21 問題解決畫面	53

一、緒論

1.1 研究背景及動機

隨著科技的進步與發展，IT已成為企業不可或缺的一部分，IT不僅輔助企業營運工作，更是幫助企業取得競爭優勢的利器之一。為了使用IT來提昇企業的競爭優勢，企業努力地將企業的IT目標與企業商業活動緊密結合。以企業的角度來看，企業訂定各項組織策略目標，然後藉由實施各項商業流程，來達到組織目標，為了讓IT與商業活動緊密結合，企業依據這些商業流程需求，提供各種IT服務。IT服務管理(Information Technology Service Management, ITSM)，跳脫傳統的思維，以服務導向的觀點，將IT視為一種服務，以服務管理的概念管理企業內部的IT，藉由提供高品質的IT服務，提昇企業的競爭優勢。

ITIL(Information Technology Infrastructure Library, ITIL)[25]提供了一套IT服務管理的實務典範，以流程導向的觀點，制定IT服務管理的流程與步驟，以及該注意的要點與方針，供企業導入IT服務管理之參考，以減少和避免錯誤的發生。ITIL將服務管理分為兩大類，分別為服務支援(Service Support)與服務傳遞(Service Delivery)，並透過服務台(Service Desk)執行各項服務管理工作。服務支援強調IT服務的日常營運管理，目的在於讓使用者能夠順利的使用IT服務來滿足他的商業功能，包含了事件管理(Incident Management)、問題管理(Problem Management)、異動管理(Change Management)、上線管理(Release Management)、組態管理(Configuration Management)等流程。服務傳遞則是強調IT服務策略面的管理，包含了財務管理(Financial Management)、服務等級管理(Service Level Management)、IT服務永續管理(Continuity Management)、可用性管理(Availability Management)、負載管理(Capacity Management)等流程。在服務支援管理中，事件管理是最重要且最根本的基礎，透過良好的事件管理，可以在發生新事件時，迅速地將事件解決或者提供暫時性解決方案(Workaround)給受影響的使用者，降低對業務運作的負面影響。

ITIL定義的事件管理流程中，企業需要針對任何不屬於IT服務的標準操作程序，可能引起資訊服務中斷或降低品質的突發事件，進行事件記錄、分類、調查與診斷、解決以及結案等事件管理流程的動作，服務台人員需要記錄事件的相關資訊，以及掌握有關的組態資訊，將事件作正確的分類，然後採取適當的措施，迅速的將服務回復到標準操

作狀態。在ITIL定義的事件管理流程中，只針對流程與步驟，以及應該注意的方針及要點，提出實務指引，並未透過很好的知識支援，協助進行事件管理。知識支援之功能可以在事件處理時，提供正確且及時的事件知識支援，協助快速的解決新事件，讓事件管理的效果顯著提升。因此必須提供很好的知識支援系統，來協助企業快速且有效的進行事件管理。

此外，事件通常有不同的情境(Context)，情境資訊的取得，可以協助處理人員掌握較完整的事件資訊，事件的情境也會影響我們對事件的處理與解決。情境相關的研究通常於情境認知(Context-aware)的普及運算(Pervasive Computing) [8][9][10][11]的建置與應用，探討在行動運算的環境中，使用各項情境認知技術，自動地適應當下的情境，提供情境認知的服務。此外，有些研究則是探討以情境資訊為基礎的支援系統[4][16]或是情境認知的問題解決[3]，說明依據使用者當下情境，提供適當的資訊以及決策支援。雖然目前情境相關的研究都應用在其他領域，並無針對IT服務管理的情境相關研究，但是情境的概念可以運用於IT服務管理，透過情境認知的方式，在適當的情境時採取適當的管理協助。

因此，本研究希望提出情境認知的事件管理知識支援架構，協助服務台快速且正確的掌握事件相關資訊以及組態資訊，並提供適當的事件管理知識支援，協助服務台人員快速地解決新事件。我們希望擷取事件的情境資訊，情境資訊包含使用者、時間、地點以及系統等，透過情境資訊的協助，使用者及服務台人員可以掌握豐富的事件相關資訊，也能根據情境資訊當作指引(Index)，找到相關的組態資訊。此外，我們採用資料探勘的技術，希望從事件歷史資料中，找出有用的知識。資料探勘強調能從大量的資料中，探勘有趣且隱藏的有用知識，供使用者作為決策參考的資料。透過資料探勘的技術，我們可以從事件管理的歷史資料中，找出情境與事件處理之間的有效知識，找出在不同情境下，對於事件的有效知識。然後透過情境認知的方式，在事件處理流程中，根據情境資訊，推薦適當的事件管理知識支援。

1.2 研究目的

根據以上的研究背景與動機，本研究希望達到下列研究目的：

- 建構IT服務環境中的情境模型，找出IT服務管理中事件管理流程活動所需的相關情境資訊內容。根據建造的情境模型，協助事件發生時，擷取適當的事件情境資訊。並透過情境資訊的指引，找到相關的組態資訊，藉此提供事件管理所需的詳細事件資訊以及相關組態資訊。
- 使用資料探勘(Data Mining)的技術，從事件歷史紀錄中，找出情境資訊與事件處理資訊有關的知識。本文針對事件調查與診斷以及事件解決兩個活動，探勘情境資訊與其之間的知識，在未來處理新事件時，系統透過推論的方法，應用上述探勘出的知識規則，依據事件擷取的情境資訊，推論得知最可能的事件知識，自動提供情境認知的知識推薦。
- 利用Web介面的瀏覽方式，建置一維型系統，提供情境認知的事件管理知識支援，協助服務台人員快速的解決新事件。



1.3 論文架構

本論文共分為六章，第一章為「緒論」，對研究動機、背景、目的以及論文整體架構作一簡單的說明。第二章為「文獻探討」，介紹與本研究相關的主題。研究主題包含，情境與情境認知、ITIL 與事件管理，以及資料探勘等。第三章為「情境認知的事件知識支援架構全覽」，介紹本架構的主要目的，以及概述各模組的主要功能。第四章為「情境認知的事件知識支援」，詳述架構中各模組的功能及其設計方法。第五章為「系統建置與開發」，說明雛型系統實作的方式與流程、開發工具與平台、與建置出的情境認知事件知識支援雛型系統功能。第六章為「結論與未來工作」，總結本研究的研究成果，並針對未來可能的研究方向加以說明。



二、文獻探討

本研究欲在 IT 服務管理的環境中，設計一個情境認知的知識支援架構，應用於事件管理流程，讓管理事件的服務台人員可以快速的解決使用者的新事件，提升 IT 服務的服務品質。本章針對相關研究主題進行探討：

2.1 ITIL (Information Technology Infrastructure Library, ITIL)

ITIL 是由英國組織OGC (Office of Government Commerce) [25]於1980年所提出，是一套IT服務管理的方法論，提供一系列由政府組織或私人機構所得IT服務管理的最佳化實務(Best Practice)。ITIL 包含一系列的叢書，書的重點在於提供有關的指引(Guidance)，包含如何提供有品質的IT服務，以及有關支援IT時所需要的功能和環境需求等指引。

2.1.1 服務管理 (Service Management)

IT服務管理著重於組織的商業需求有關的IT服務，對這些IT服務進行傳遞與支援的管理。因此在ITIL的服務管理中，分成服務支援 (Service Support)[18]及服務傳遞 (Service Delivery)[19]。服務傳遞管理重點在於，為了提供企業客戶適當的支持，企業提供者的商業需求需要提供哪些服務，為了提供這樣的支援，定義了財務管理(Financial Management)、服務等級管理(Service Level Management)、IT服務永續管理(IT Service Continuity Management)、可用性管理(Availability Management)、負載管理(Capacity Management)等流程，來協助服務傳遞管理。服務支援管理重點在於，保證使用者能夠使用適當的服務來滿足商業功能(Business Function)，為了提供這樣的支援，定義了事件管理 (Incident Management)、問題管理 (Problem Management)、異動管理 (Change Management)、上線管理 (Release Management)、組態管理 (Configuration Management) 等流程，來協助服務支援管理。最後再加上服務台 (Service Desk) 的功能，服務台是IT服務提供者提供給客戶的單一連絡窗口，藉由這一個單一窗口，讓使用者能夠與服務提供者聯繫。

由於IT服務管理範疇的廣大，在本論文中，將只針對服務支援中的事件管理流程作深入研究，下面，我們對事件管理流程作詳細的說明。

2.1.2 事件管理 (Incident Management)

在ITIL中，定義事件(Incident)為，「任何不屬於服務之標準操作程序的一部份，因而導致服務中斷或是減低服務的品質，這樣的情況發生就稱為事件」。因此，當使用者遭遇以上的狀況時，就會向服務台引發事件請求，服務台需要採取適當的處理動作，而這些動作就稱為「事件管理」。而一系列事件管理的活動，就稱為事件管理流程。在ITIL中，定義事件管理流程為，「當事件發生直到服務回到標準的作業，對於事件辨別、紀錄、分類以及事件處理進度紀錄等活動，稱為事件管理流程」。目的在於解決事件，盡快的將IT服務回復到標準的操作環境。事件管理的活動，包含下列幾項活動：

- 事件偵測與紀錄(Incident detection and recording)

從服務台或是系統異常狀況管理系統偵測所得到的事件細節資料，是事件管理的輸入(Input)。在這個活動中，需要記錄事件基本的細節資料，然後依據狀況，看是否向專業支援小組發出警報。如果是個預設的服務請求(Service Request)，則是啟動對應的程序(Procedure)來處理這個服務請求。

- 事件分類與初始支援(classification and initial support)

根據紀錄的事件細節，以及從組態管理資料庫(Configuration Management DataBase, CMDB)所得到的組態資訊，進行事件的分類，事件分類是根據事件的起源或者是症狀，將事件辨別為某個類別的事件。另外針對事件資訊，決定緊急程度以及影響程度，進而判斷事件處理的優先程度。最後根據事件的資訊，提供初始的支援，提供快速的解決方式。

- 調查及診斷(investigation and diagnosis)

在這個階段所要進行的動作有，仔細的評估事件的細節，收集以及分析所有相關的資訊，以及提出解決方案，或者是將事件安排給後端的專業人員。

- 解決與恢復(resolution and recovery)

在這個階段所要進行的動作為，採取解決方案或是暫時性解決方案(Workaround)給事件發起者，或者引發一個異動請求(Request For Change, RFC)來解決這個事件。如果採取的解決方法無法正確的解決事件，要採取回復的動作，回復原本的狀態。

- 事件結案(Incident closure)

當事件解決後，要得到顧客或者是原本事件的提出者的確認訊息，確定事件有正確被解決，才可以將事件結案。

- 所有權、監控與溝通(ownership, monitoring, tracking and communication)

這個階段所要進行的動作有，監看事件處理情況，當事件無法處理，即將違背服務等級時，要進行事件向上提報(Escalating)的動作，最後，還要通知使用者事件的狀況。

在本論文中，也將針對 ITIL 中的事件管理流程，提供情境認知的知識支援，支援流程中的各個活動，並主要專注於事件分類與初始支援、調查與診斷、解決與回復等三個階段，提出情境認知對於這些階段的協助。

2.2 情境與情境認知 (Context and Context-aware)

2.2.1 情境(Context)

根據Schilit and Theimer[12]的定義，情境是「描述鄰近地點的人與物件的地點與身份識別」。Dey等人定義情境[5][6]，「任何可以描述一個個體狀況的資訊，這邊指的個體可以是人員、地點或是當使用者使用服務時，所有有關的物件，物件可能為使用者、服務或是提供這個服務的元件等」。Dey[7]等人將情境著重在描述使用者的實體環境、社交狀況、情緒狀況以及資訊的狀況。Schilit等人認為情境的重要性在於描述持續變化的環境資訊，記錄了你的位置、身邊有誰，以及你身邊有哪些資源。因此，在他們的架構

中[8]，情境包含下列部分的環境：

- 運算環境(Computing environment)：使用者可以使用的計算機、可以登入以及展示資訊的設備、網路頻寬、網路連線狀況等。
- 使用者環境(User environment)：使用者的位置、鄰近使用者的狀況以及使用者之間的社交狀況(Social situation)。
- 實體環境(Physical environment)：燈光以及噪音程度等。

在本論文中，定義情境為，「任何可以描述一個個體狀況的資訊，這邊指的個體可以是人員、地點、時間，以及當使用者使用IT服務時，所有有關的物件，物件可能為使用者、服務提供者、IT服務本身以及提供IT服務的元件」。並且，本論文的情境，著重於描述使用者在使用IT服務提供者所提供的IT服務時，相關環境資訊，包含了以下部分的環境：

- 使用者環境：使用者的個人資訊與組織資訊、使用者的位置、鄰近使用者的狀況、使用者之間的社交狀況(Social situation)，使用者當下的活動。
- IT服務環境：使用者可以使用的IT服務、用來提供使用者IT服務的系統元件，系統元件的狀態，以及IT服務的提供者與支援者的資訊。

2.2.2情境分類 (Categories of Context)

適當的情境分類，可以協助設計者在設計他們的應用程式時，對於情境有適當的應用。Ryan等人[12]將情境分類為所在位置(Location)、環境(Environment)、身分識別(Identity)與時間(Time)。Schilit[13]等人則是將情境分類為所在位置(Location)與身份識別(Identity)。Dey[5][7]等人將情境分類為所在位置(Location)、活動(Activity)、身分識別(Identity)與時間(Time)，用這四種情境類別的資訊，來描述特定個體目前的狀態。在本論文中，將情境分類分為所在位置(Location)、活動(Activity)、身分識別(Identity)以及時間(Time)，利用此四類的情境分類，來描述IT服務環境中特定個體的狀態資訊，其中，針對提供IT服務的系統元件，則是使用系統狀態(Status)的情境分類來代替活動(Activity)的情境分類。

2.2.3 情境認知 (Context-aware)

情境認知最早是由Schilit與Theimer在1994年提出[13]，針對軟體(software)提出情境認知的應用，定義情境認知為「適應軟體使用的所在位置、附近的人員及物件，以及這些物件根據時間所做的改變」，也就是說，能夠依據環境中的使用者、時間、地點、物件以及隨著時間的變化，適應環境中的變化，而採取相對應的動作。Dey等人定義情境認知為[5][7]，「一個情境認知的系統，要能適應情境來提供使用者有關的資訊以及服務，而有關性則是根據使用者的任務做判斷」。

由以上得知，情境認知最重要的就是要能夠「適應情境」，採取對應的動作。在本論文中，採取Dey等人的定義，定義情境認知為「一個情境認知的系統，要能使用情境來提供使用者有關的資訊以及服務，而有關性則是根據使用者的任務做判斷」，另外，本論文目的在提供情境認知的事件管理協助，因此，使用者的任務為事件管理的活動，針對事件管理活動提供有關的資訊及服務。

2.3 資料探勘(Data mining)

資料探勘[17]主要目的在於從大量資料中，挖掘出隱藏的有用知識，以提供決策者做參考指標，常見的資料探勘研究有關聯規則(association rules)[1][2]、分類法與預測(classification and prediction)、分群分析(cluster analysis)等。下面，我們針對本論文中所使用的資料探勘技術，關聯規則，做深入的介紹。

2.3.1 關聯規則探勘(association rule mining)

關聯規則探勘目的在從給定的資料集中，找出資料欄位之間有趣的關連性。關聯規則的基本概念如下，假設 $J = \{i_1, i_2, \dots, i_m\}$ 是一群項目(item)的集合， D 是任務相關(task-relevant)的資料，由一群交易(transaction)資料所組成，而每一筆交易 T 包含了一群項目所組成，其中 $T \subseteq J$ 。假設每筆交易有一個表頭(identifier)，稱作TID。此時我們假設 A 是一組項目的集合。此時，若有一筆交易宣稱它包含 A ，則須滿足且只滿足 $A \subseteq T$ 。

而我們將關聯規則用以下的形式表示：

$$A \Rightarrow B \quad \text{where } A \subset J, B \subset J, A \cap B = \phi$$

假設在D的集合中，存在s% 的交易包含 $A \cup B$ ，則我們稱 $A \Rightarrow B$ 的支持度(support)為s，然後以機率的方式表示為 $P(A \cup B)$ 。假設在D的集合中，存在c% 的交易具備存在A之後剛好又存在B的交易，則我們稱 $A \Rightarrow B$ 的可信度(confidence)為c，以條件機率的方式表示為 $P(B|A)$ 。表示如下：

$$\text{Support}(A \Rightarrow B) = P(A \cup B) \quad (2.1)$$

$$\text{Confidence}(A \Rightarrow B) = P(B|A) \quad (2.2)$$

關聯規則探勘可以分成下面兩個執行步驟：

(1). 找出所有頻繁項目集(frequent itemsets)。

根據設定好的最小支持度，我們可以找到支持度大於最小支持度的項目集。

(2). 從頻繁項目集中產出關聯規則。

同樣的，這些產生出來的關聯規則，必須符合最小支持度與最小可信度。

2.3.2 以限制為基礎的關聯規則探勘(Constraint-based association rule mining)

在給定的資料集中，根據資料探勘產生的結果，可能會產生數量龐大的規則，而這些規則中，有很多是使用者不感興趣的。因此，在限制為基礎的關聯規則探勘中，探勘的內容是根據使用者所提供的限制條件為指示(Guidance)，進行資料探勘，如此一來，將對得到使用者感興趣的規則。限制條件有下列幾種形式：

- 知識型態(Knowledge Type)限制：規範需要探勘出哪種類型的知識，例如關聯型(association)。
- 資料限制：規範要找出哪些有相關的資料集進行資料探勘。
- 維度(Dimension)或等級(Level)限制：規範要使用哪個維度的資料，或者是概念階層中那個等級的資料。

- 規則(Rule)限制：規範被探勘規則的形式。這種的限制通常會表示成 meta-rules，也就是規則樣版(rule templates)，例如限制述語(predicate)於規則的前項(antecedent)或後項(Consequent)出現的最大次數或最小次數，或者限制資料屬性、資料屬性間關係和資料聚集(aggregate)等。

在本論文中，採用以限制為基礎的關聯規則探勘，並採取規則限制的方法，限制出現於規則的前項與後項中述語的資料屬性，然後進行關聯規則探勘，找出我們感興趣的事件管理知識樣版。

2.4 本體論(Ontology)

本體論(Ontology)源自於哲學，用來探討「存在(existence)」的這個主題，通常利用樹狀結構以及關聯的方式來表示各種事物，並描述這些事物之間的規則。本體論也被用於人工智慧(A.I.)的領域，用於知識分享或是知識工程中，去描述知識構成的要素，以及要素之間的關係。Truber[10]將本體論應用於知識分享上，定義本體論為「概念化的一個形式的規則說明(a specification of a conceptualization)」。Uschold[14]則是將本體論解釋為「本體論是一個正式的(formal)且明確的規格說明，旨在說明大家都能共同接受的概念」。根據W3C的定義，本體論用來描述與表示各種領域的知識，本體論可以被入、資料庫或應用軟體用來分享專業領域的資訊。在此我們使用本體論來描述情境資訊，我們訂定各個情境特徵本體，描述環境中個體的情境資訊。

2.5 OWL

OWL(Web Ontology Language)[20]是W3C推薦的本體論描述語言，由DAML 和 OIL[24]合併而來。OWL 是為了讓應用程式可以處理文件中所包含的訊息而設計，有別於將內容呈現給人類了解。OWL包含三種子語言，OWL Lite、OWL DL、OWL Full。

OWL Lite

OWL Lite 適用於只需要簡單的類別階層以及簡單的限制的使用者。舉例來說，它支援基數(Cardinality)限制，只允許基數值為 0 或 1。

OWL DL

OWL DL 適用於需要最大限度描述能力的使用者，支援進行運算的使用者，保證計算的完全性(Computational completeness)以及計算的可決定性(Decidability)。

OWL Full

OWL Full 不僅提供最大限度的描述能力，並且提供使用 RDF 來增加語法的自由性，但是並非所有的推理軟體可以完全支援 OWL Full 的推論，因此並不提供運算的保證。

因此此三種語言之間的關係如下所示，反之則不成立。

- 每一個合法的 OWL Lite 本體都是一個合法的 OWL DL 本體。
- 每一個合法的 OWL DL 本體都是一個合法的 OWL Full 本體。
- 每一個有效的 OWL Lite 結論都是一個有效的 OWL DL 結論。
- 每一個有效的 OWL DL 結論都是一個有效的 OWL Full 結論。



三、情境認知事件知識支援架構

本研究提出一個情境認知事件知識支援架構，在 IT 服務管理的環境下，透過事件發生時情境資訊(Context Information)的收集、處理、推論與散播，與知識發現(Knowledge Discovery)技術從事件記錄 (Incident Log) 挖掘相關的事件管理經驗知識，依據事件發生時的情境資訊，使用規則推論得到有用的知識樣板，提供服務台情境認知(Context-aware)的事件管理協助，快速且有效辨別新事件與找到適合的解決方案。

為了達到上述目的，我們提出的架構，如下圖 3-1 所示。情境模組從事件發起者環境中偵測與擷取事件情境資訊，透過情境呈現模式 (Context Modeling)，以結構化的方式解釋與呈現情境資訊。資料探勘模組從歷史事件記錄，挖掘隱含且相關的事件處理知識，這裡主要探討事件辨別與事件解決方案跟事件情境資訊之間的關係。推論模組使用規則式推論，協助本架構中各項推論工作，包含情境推論與事件處理支援時的推論。情境認知事件支援模組負責支援事件處理流程的活動，根據事件的情境狀況，提供有效的資源，分為情境認知的事件辨別知識支援、情境認知的事件解決知識支援與情境認知的事件相依知識支援。各個模組的功能，簡述如下：

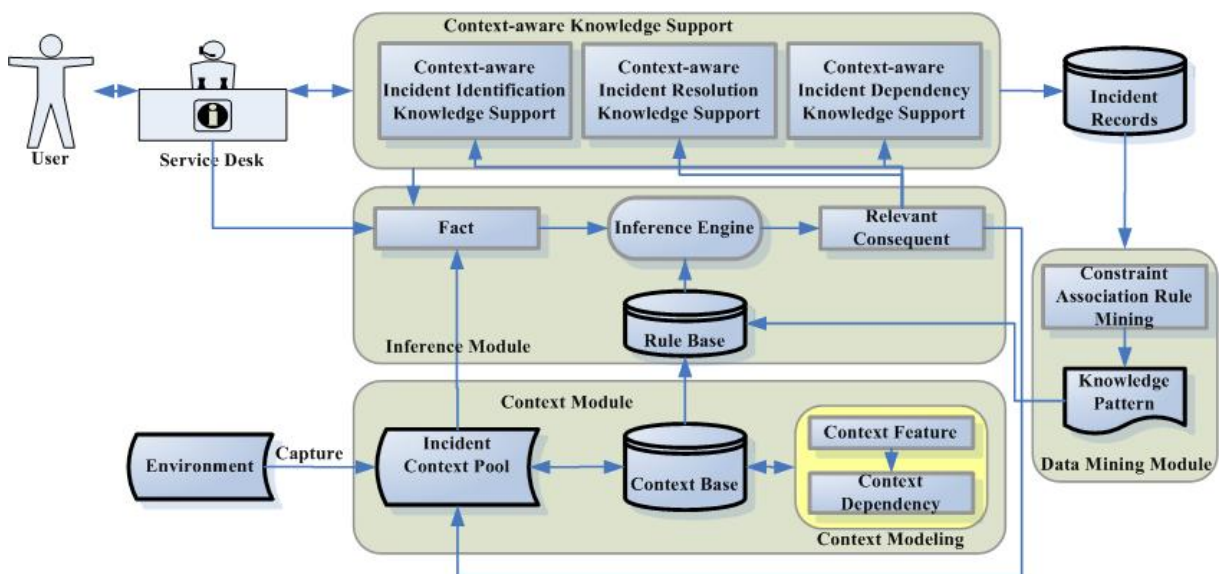


圖 3.1 架構全覽

Context Module 負責情境資訊呈現，使用OWL(Web Ontology Language)[20]語言描述與呈現情境資訊，達到情境資訊共享、重複使用與可讀性等目的。另外，當事件發生時，負責擷取與紀錄事件發生時的完整事件情境資訊供其他模組使用。

Data Mining Module 負責從事件紀錄中，使用資料探勘中關聯式規則的技術，找出情境與事件辨別、情境與解決方案與事件間關聯性的知識樣板(Knowledge Pattern)。

Inference Module 負責使用規則式推論方法協助本架構中各項推論工作。包含事件情境推論，推論事件發生時完整的情境資訊；情境相關症狀推論，推論事件可能症狀描述；情境相關解決方法推論，推論適合的事件解決方法。

Context-aware Knowledge Support Module 負責在事件管理流程的活動中依據當下的情境，採取適當的協助，這裡主要探討情境認知的事件辨別知識支援、情境認知的事件解決知識支援與情境認知的事件相依知識支援。



四、情境認知事件知識支援模組介紹

下面，我們針對各模組做詳細的介紹：

4.1 情境模組(Context Module)

在這個模組中，主要有兩個目的，第一是情境資訊結構化的組織與呈現 (Context Modeling)，在此我們採用本體論 (Ontology) 與 OWL 語言來描述情境資訊。第二是擷取事件發生時的情境資訊儲存於事件情境池(Incident Context Pool)，將事件發生時的情境擷取與收集儲存，供本架構中其他模組使用。

情境模組，如圖 4.1 所示，Context Modeling 負責使用 OWL 語言描述情境資訊，然後存放於情境庫中。另外，從環境中擷取的資訊將會放置於事件情境池中，並且根據得到的情境，透過推論模組推論，得知無法經由偵測的情境資訊，然後放置於事件情境池中。

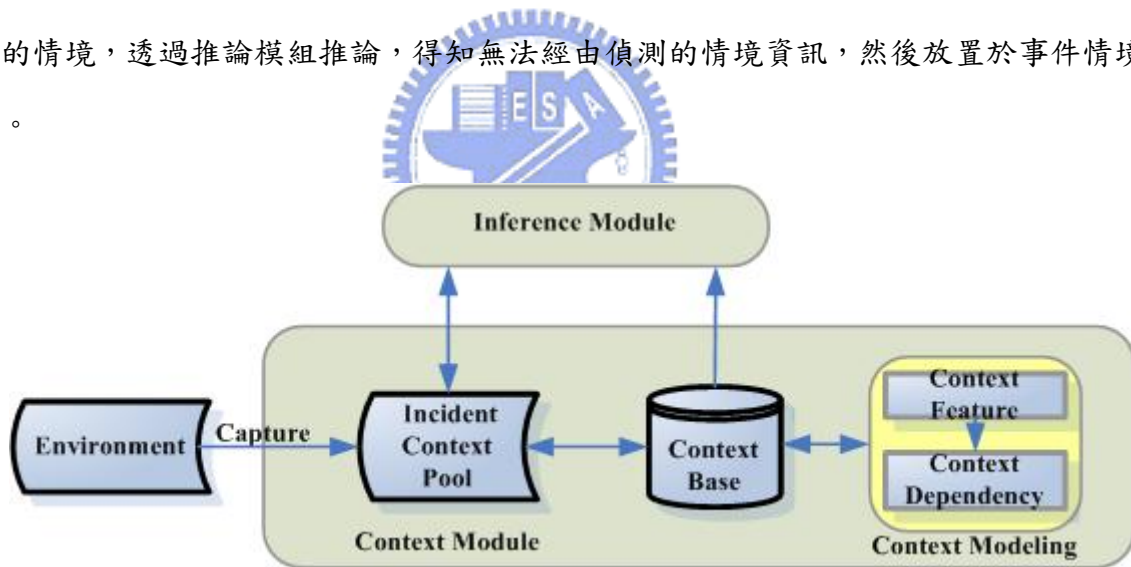


圖 4.1 情境模組架構圖

4.1.1 情境模式化(Context Modeling)

任何可以描述一個個體狀況的資訊，都稱為情境資訊，這裡的個體可以是人員、地點或是使用者使用服務時有關的物件，物件可能為服務使用者、服務本身或者是提供這個服務的元件等。由於情境資訊的多樣性，要完整的將這些情境資訊表示出來，是不切實際的。因此，我們針對 IT 服務管理的環境，並採用三層式的描述方法來設計我們的

情境資訊，上層是所有情境概念的抽象化，而越往下層對情境描述則越為詳細。其中最上層為情境個體層(Context Entity Layer)，我們根據 IT 服務管理的精神，訂定了情境個體，包含個人、服務以及系統，第二層為情境特徵層(Context Feature Layer)，定義環境中個體的情境特徵，例如人員、系統、時間與地點等情境特徵，一個情境個體可能包含一個或多個情境特徵，例如情境個體「服務」可能包含情境特徵「個人」，描述服務提供者的資訊、顧客的資訊，以及情境特徵「服務」，描述服務本身相關的資訊，例如服務名稱、服務的參數等。而下層則為情境相依層(Context Dependency Layer)，用來描述情境個體之間的相互關係，例如人與系統的關係，可能是某個人使用某個系統或者是某個人負責管理某個系統，用來找到無法經由偵測的情境資訊以及描述更完整的情境資訊。藉由這樣三層的描述方法，描述 IT 服務管理環境下的情境資訊。

4.1.1.1 情境特徵層 (Context Feature Layer)

根據 IT 服務管理的精神，由構成 IT 服務的主要元件，人、服務、系統、時間與地點等，做為我們主要描述的情境個體，依據這些情境個體，我們定義了下列幾項的情境特徵，如下圖 4.2 所示：

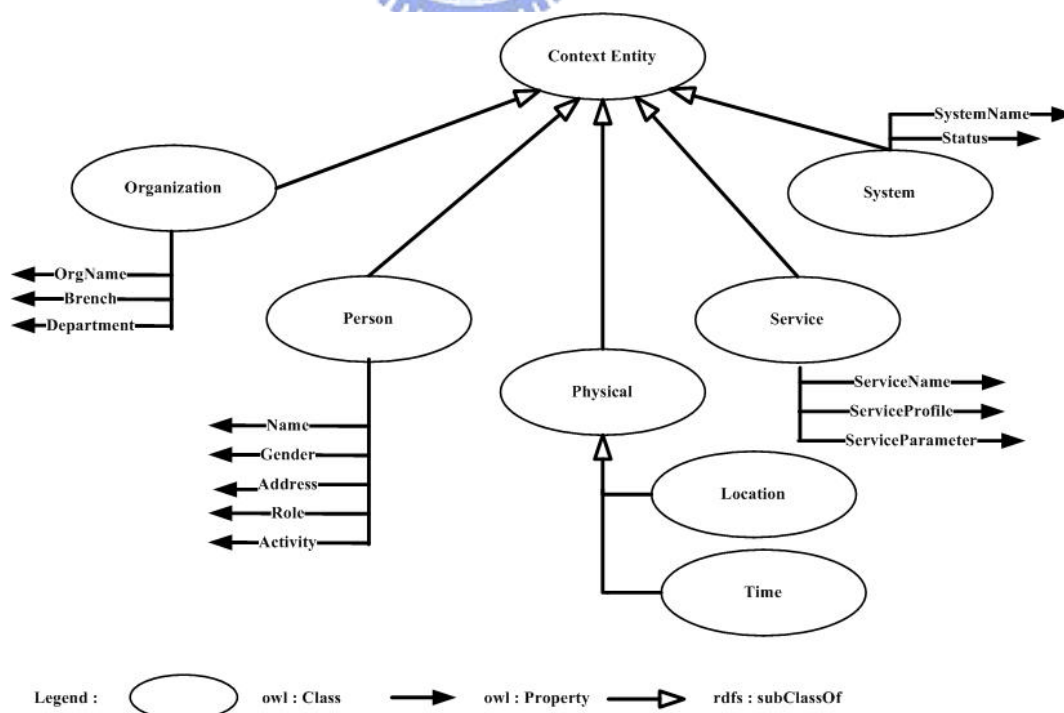


圖 4.2 情境特徵本體 (Context Feature Ontologies)

情境個體由情境特徵描述，情境特徵包含 Person、Organization、Service、System 以及 Location 與 Time，一個情境個體可能包含一個或多個情境特徵，例如描述個體“Mary”，就包含了“Person”、“Organization”與“Location”等情境特徵。下面我們將對各個情境特徵做詳細的說明：

➤ 實體類(Physical)

描述情境個體的實體環境資訊，包含時間 (Time)、地點 (Location)。

➤ 人員(Person)

描述情境個體中個人的資訊，包含用來做身分識別(Identity)的名字、職務角色、性別與住址等。

➤ 服務(Service)

描述情境個體中服務的資訊，包含服務名稱、服務概況(Profile)與服務參數等。

➤ 系統(System)

描述情境個體中系統的資訊，包含系統名稱、系統描述與系統狀態等。

表 4.1 情境特徵例子

Location
<pre><Person rdf:ID="Mary"> <Located_In> <Floor2 rdf:ID="MB208"/> </Located_In> </Person></pre>
Interpretation：人員“Mary”，目前所在地點為“MB208”。
Person
<pre><Person rdf:ID="Mary"> <Sex> Female </Sex> <Age> 25 </Age> <PhoneNumber>57424 </PhoneNumber> <Role> Employee </Role> </Person></pre>
Interpretation：人員“Mary”，性別為“女性”，年齡為“25”歲，電話號碼為“57424”以及職務角色為“員工”。
Service
<pre><Service rdf:ID="Printing"> <Description> all about printing </Description> </Service></pre>
Interpretation：服務名稱為“列印服務”，服務敘述為“所有跟印表有關的事務”。
System
<pre><PC rdf:ID="PC071"> <CPU> P4-2.0G </CPU> <Status> normal </Status> </PC></pre>
Interpretation：系統名稱為“PC071”，包含的屬性，CPU 為“P4-2.0 G”，記憶體為“DDR 512M，以及狀態為“正常”。

4.1.1.2 情境相依層(Context Dependency Layer)

情境相依負責描述情境個體之間的關係，我們定義了許多不同的情境相依關係，如下圖 4.3 所示，每個情境特徵除了具有自己的屬性以外，還具備與其他情境特徵之間的相依關係(Dependency)，藉此得到一些無法經由偵測得到的情境資訊。下面，我們將針對情境相依關係作詳細的說明。

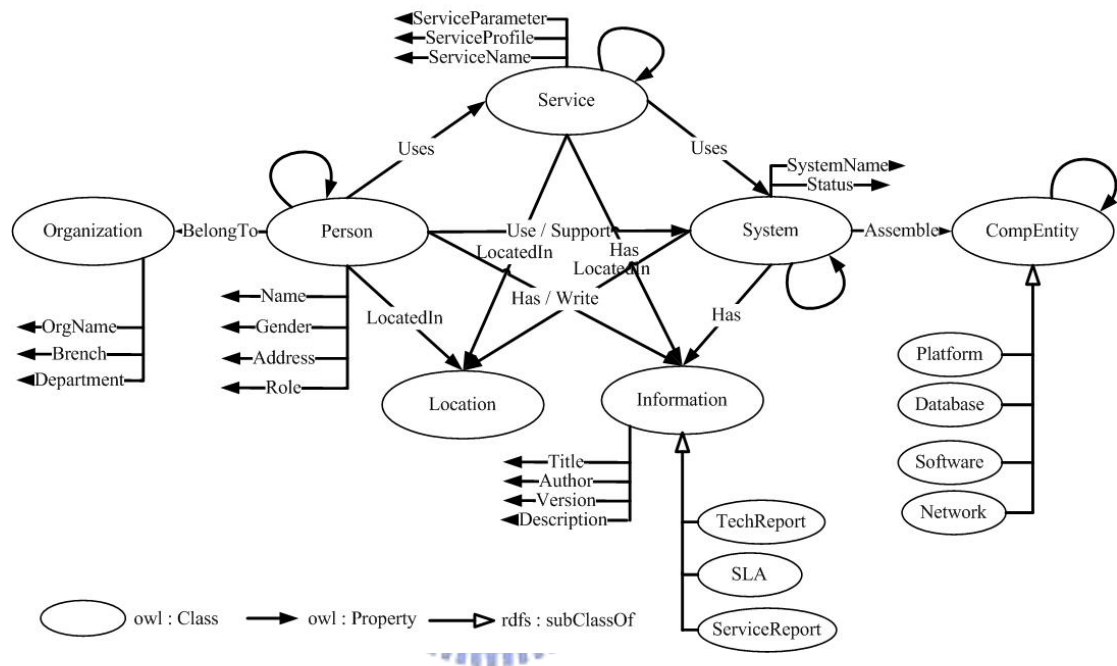


圖 4.3 情境相依關係 Context Dependency

➤ **Organization** Dependency

描述組織個體與人員之間的關係。例如，人員“Mary”隸屬於組織的“RD”部門。

➤ **System** Dependency

描述系統個體以及系統元件之間的關係。例如，Assemble(System_A1, PC01)表示系統“PC01”是構成系統“System_A1”的一部分。

➤ **Service** Dependency

描述不同服務個體之間的關係。例如，“E-Mail”服務通常與“Internet”服務有關係。

➤ **Person** — **Service** Dependency

描述組織個體與服務個體之間的關係。例如，“會計部門”使用“Accounting”服務。

➤ **Person — System Dependency**

描述組織與系統個體之間的關係。例如，“會計部門”使用系統“System_A1”。

➤ **Service — System Dependency**

描述服務個體與系統個體之間的關係，也就是描述 IT 服務提供者在提供服務時，為了支援服務所使用的系統元件。例如，“E-Mail”服務是由系統“System_A1”所提供的。

➤ **Service — Information Dependency**

描述服務元件與資訊之間的關係。例如，“Email”服務與其服務合約(Contract)文件之間的關係。

➤ **System — Information Dependency**

描述系統個體與資訊之間的關係。例如，“System_A1”與其“技術文件”之間的關係。



表 4.2 情境相依關係例子

<p>Organization Dependency</p> <pre><owl:Class rdf:ID="DepartmentAccounting"> <rdfs:subClassOf> <owl:Class rdf:about="#BranchTaipei"/> </rdfs:subClassOf> </owl:Class></pre>
<p>Interpretation : 會計部門隸屬於台北分公司。</p>
<p>System Dependency</p> <pre><System rdf:ID="System_A1"> <Assemble> < Sys_Com rdf:ID="Router_A1"/> < Sys_Com rdf:ID="Switch_A1"/> < Sys_Com rdf:ID="Server_A1"/> </Assemble> </System></pre>
<p>Interpretation : 系統"System_A1"是由系統元件"Router_A1", "Switch_A1"以及"Server_A1"所組成的。</p>
<p>Service Dependency</p> <pre><Service rdf:ID="AccountingService"> <LinkTo> < Service rdf:ID="ProcurementService"/> </LinkTo> </Service></pre>
<p>Interpretation : "會計"服務與"採購"服務具有"LinkTo"的關係。</p>
<p>Organization — Service Dependency</p> <pre><Organization rdf:ID="RD Dept."> <Use> < Service rdf:ID="CAD Service "/> </Use> </Organization></pre>
<p>Interpretation : 組織中"RD Dept.", "使用" "CAD"服務。</p>
<p>Service — System Dependency</p> <pre><Service rdf:ID="AccountingService"> <Use> < System rdf:ID="Server_A2"/> </User> </Service></pre>
<p>Interpretation : "會計服務" "使用"系統"Server_A2"。</p>
<p>Location Dependency</p> <pre><Person rdf:ID="Mary"> <LocatedIn> < System rdf:ID="MB205"/> </LocatedIn> </Person></pre>
<p>Interpretation : 人員"Mary"所在地點為"MB205"。</p>

4.1.2 事件情境資訊池(Incident Context Pool)

當事件發生時，我們擷取其情境資訊，協助進行事件管理。當系統偵測新事件發生時，擷取環境中相關個體的情境特徵，再將這些資訊當作輸入值 (Input)，透過推論模組進行推論，推論得到跟此事件有關的情境相依資訊，藉此得到這件事件的情境資訊，然後將這些資訊儲存於事件情境池，提供其他模組使用。下面用一個例子說明，如表 4.3 所示，使用者“PTC”遭遇“E-mail”服務無法使用的問題，此時的情境資訊包含人員“PTC”的相關資訊、服務“E-mail”的相關資訊、系統“SystemA”的相關資訊以及“PTC”的地點資訊，構成事件的情境資訊。

表 4.3 事件情境關係池範例

```
<Person rdf:ID="PTC">
  <Gender> Male </Gender>
  <Age> 25 </Age>
  <Role> Employee </Role>
  <Belong_To>
    <Organization rdf:ID="PTC_Company">
      <Department> DepartmentRD </Department>
    </Organization>
  </Belong_To>
  <Uses>
    <Service rdf:ID="EMailService">
      <Has_SLA>
        <GoldSLA> GoldServiceLevel </GoldSLA>
      </HasSLA>
      <Has_System>
        <System rdf:ID="SystemA">
          <System_Status> ServerProblem </System_Status>
          <Aggregate_Of>
            <Server rdf:ID="ServerA"/>
              <Status> Abnormal </ Status>
            <Router rdf:ID="RouterA"/>
              <Status> Normal </ Status>
            <Switch rdf:ID="SwitchA1"/>
              <Status> Normal </ Status>
            <PC rdf:ID="PC005"/>
              <Status> Normal </ Status>
          </Aggregate_Of>
        </System>
      </Has_System>
    </Service>
  </Uses>
  <Located>
    <HsinChu rdf:ID="S505"/>
  </Located>
</Person>
```

4.2 資料探勘模組(Data Mining Module)

資料探勘模組，如圖 4.4 所示，負責從歷史事件紀錄中，使用限制式關聯規則探勘的技術，找出有用的事件情境與事件處理相關的知識樣板(Knowledge Pattern)，再將這些知識樣板傳送給推論模組，供推論模組使用。下面，我們將對於模組中的其他部份進行深入的討論。

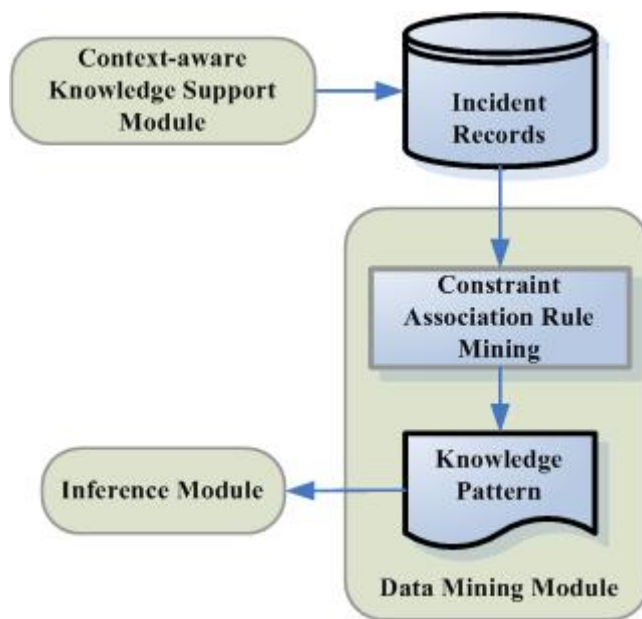


圖 4.4 資料探勘模組架構(Data Mining Module)

4.2.1 事件報告內容(Overall of Incident Report)

事件報告記錄了事件的完整資訊，也是我們進行資料探勘的資料來源，一筆事件報告所包含的資料，如表 4.4 所示，資料欄位包含事件編號(Incident_ID)、事件發生時間(Time)、事件發生地點(Location)、使用者姓名 (Creator_Name)、使用者當時活動(C_Activity)、使用者部門(C_Dept)、服務名稱(Service Name)、服務等級(Service Level)、系統名稱(System Name)、系統狀態(System Status)、事件影響程度(Impact)、事件緊急程度(Urgency)、事件處理優先程度(Priority)、使用者對於事件的描述(User Description)、系統錯誤訊息(System Error Message)、事件類別(Incident Category)、事件描述(Incident Description)、解決員工姓名(Supporter)、解決員工部門(S_dept.)、解決方案(Resolution Description)以及解決時間(Resolving_time)。

表 4.4 事件報告內容

ID	INCool	Time	4/20 AM11:00
Creator	PTC	C_Activity	meeting
C_dept	RD	Location	R505, HS
Service	E-Mail	SLA	Gold
User Description	Virus detected.	System_Error_Msg.	virus
Impact	Level 1	Urgency	High
Inc_Category	Platform Failure	Priority	3
Inc_Desc.	PC virus infected.		
Resolution_Type	Expert		
Resolution_Desc.	Run antivirus program to remove virus.		
Supporter	Clay	S_dept	MIS02
Resolving_time	2005/4/21		

為了資料探勘的需求，我們將這些資料欄位分類，如表 4.5 所示，我們總共分成四個類別，分別為情境(Context)類別、使用者事件資訊(User Incident Information)類別、管理者事件分類(Staff Incident Classification)類別以及解決方案(Resolution)類別。情境類別的資料有使用者情境，包含使用者姓名(Creator_Name)、使用者當時活動(Activity)、使用者部門(C_Dept.)以及使用者地點(Location)；服務情境有服務名稱以及服務等級；系統情境有系統名稱以及系統狀態。使用者事件資訊類別的資訊有使用者症狀描述(User_Desc.)、系統錯誤訊息(System_Error_Msg.)、事件影響程度(Impact)、事件緊急程

度(Urgency)、事件優先程度(Priority)。管理者事件分類類別的資訊有事件類別(Incident Category)以及事件描述(Incident_Desc.)。解決方案類別的資訊則有解決人員姓名(Staff_Name)、解決人員部門(Staff_dept.)、解決方案類別(Resolution_Type)以及解決方案敘述(Resolution_Desc.)。根據這樣的歷史資料，進行知識挖掘。

表 4.5 事件報告資料分類

Type	Context	User Incident Information	Staff Incident Classification	Resolution
	Time User Context Creator_Name, C_Activity User_Dept. Location Service Context Service_Name, Service Level System Context System_Name, Status	User_Desc. System_Error_Msg Impact, Urgency, Priority	Inc_Category, Inc_Desc.	Resolution type, Resolution Desc. Supporter_Name, S_Dept.

4.2.2 有限制的關聯式規則探勘(Constraint Association Rule Discovery)

我們使用有限制的關聯規則探勘(Constraint Association Rule Discovery)，來挖掘出我們所需要的知識樣版(Knowledge Pattern)。我們將探勘的知識樣板分成三種類別，如圖 4.5 所示，第一種為**情境相關的事件分類與辨別探勘**，探勘在何種情境下，常發生哪種的事件，例如得到的知識樣板為情境為使用者部門為“RD Dept.”時，常發生的事件描述為“mailbox quota exceed”的事件。第二種為**情境相關的解決方案探勘**，探勘事件發生時的情境，與解決方案之間的關係，例如得到的知識樣板為情境為使用者部門為“RD Dept.”所發生的事件描述為“mailbox quota exceed”的事件，所採取的解決方案描述為“remove mail from mailbox”的解決方案。第三種則為**情境相關的事件相依關係探勘**，探勘事件發生時，根據當時的情境，是否跟其他的事件有相依的關係，藉此可以將多件事件同時處理，或者是根據這樣的預測關係，採取預先處理的動作。例如得到的知識樣板為事件描述“PC performance Low”與事件描述“PC Crash”具有事件相依的關係，當服務台辨別事件為“PC Performance Low”的事件時，根據這樣的相依性，得知與事件“PC Crash”有相依

性，此時，服務台可以根據這樣的相依性，考慮針對此事件，採取預先處理的解決方案，也就是針對“PC Crash”的事件描述，採取對應的解決方案，避免未來發生事件描述為“PC Crash”的事件。或者是得到的知識樣板為事件描述為“can't send email”與事件描述“MailServer crash”具有相依性的關係，當服務台辨別事件為“can't send email”的事件時，依據這樣的相依性，得知與事件“MailServer crash”有相依性，此時，服務台可以根據這樣的相依性，找到目前也有事件描述為“MailServer crash”的事件，服務台就可以針對這兩個事件同時做處理。下面，我們說明如何進行資料探勘。

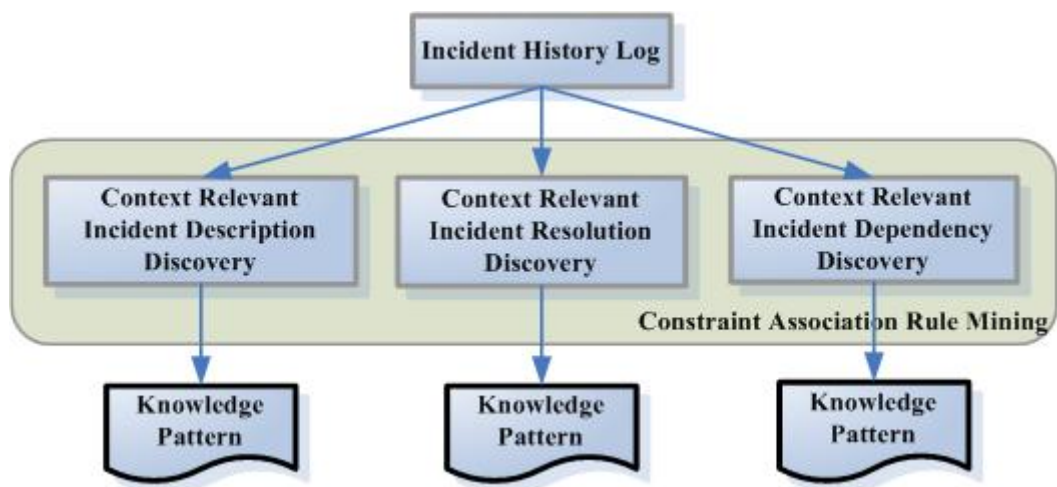


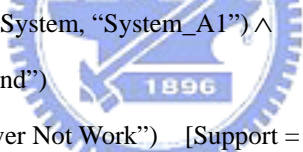
圖 4.5 限制式規則形式

4.2.2.1 Context Relevant Incident Classification Knowledge Pattern

為了探勘情境資訊與事件分類與事件描述類資訊之間的關係，我們限制規則的格式，訂定 Meta-Rule 格式如下所示：

$$\begin{aligned} & [\text{Context}(C, X)]^* \wedge [\text{User_Inc}(N, Y)]^* \rightarrow \text{Staff_Inc}(I, P) \\ & [\text{Support} = s\%, \text{Confidence} = c\%] \end{aligned} \quad (4.1)$$

根據前面的資料分類，在 Meta-Rule 格式中，規則的左手邊為情境類的資料以及使用者事件資訊類的資料，右手邊則為管理者事件分類類別中的資料，其中 **C,N,I** 代表該類別中的資料屬性名稱，而 **X,Y,P** 則是代表該資料屬性的值，符號“ \wedge ”代表 AND 的意思。此外，符號“ $*$ ”表示可能包含一個或多個資料屬性，因此在一條規則中，可能會出現一個或多個情境類別中的資料及使用者事件資訊類別的資料。如此一來，符合這樣限制的規則將會被找出來，如下所示：


$$\begin{aligned} & \text{Context}(C_Dept., \text{“RD”}) \wedge \text{Context}(\text{System}, \text{“System_A1”}) \wedge \\ & \text{User_Inc}(\text{User_Desc}, \text{“Host not Found”}) \\ & \rightarrow \text{Staff_Inc}(\text{Inc_Desc}, \text{“DNS Server Not Work”}) \quad [\text{Support} = 2\%, \text{Confidence} = 18\%] \end{aligned}$$

代表當情境為使用者部門為“研發部門”，使用系統為“System_A1”以及使用者描述為“未連到伺服器”，則有可能發生“DNS 伺服器不運作”的事件，可信度為 18%，支持度為 2%。

4.2.2.1 Context Relevant Incident Resolution Knowledge Pattern

為了探勘情境資訊與事件類資訊及解決方案之間的關係，我們限制規則的格式，訂定 Meta-Rule 格式如下所示：

$$\begin{aligned} & [\text{Context}(C, X)]^* \wedge [\text{Staff_Inc}(I, P)] \rightarrow \text{Resolution}(R, Q) \\ & [\text{Support} = s\%, \text{Confidence} = c\%] \end{aligned} \quad (4.2)$$

根據前面的資料分類，在 Meta-Rule 格式中，規則的左手邊為情境類的資料以及管理者事件分類類別的資料，右手邊則為解決方案類的資料，其中 **C,I,R** 代表該類別中的資料屬性名稱，而 **X,P,Q** 則是代表該資料屬性的值，符號 “^” 代表 AND 的意思。此外，符號 “*” 表示可能包含一個或多個資料屬性，因此在一條規則中，可能會出現一個或多個情境類別中的資料及使用者事件資訊類別的資料。如此一來，符合這樣限制的規則將會被找出來，如下所示：

$$\begin{aligned} & \text{Context}(\text{C_Dept}, \text{"RD"}) \wedge \text{Staff_Inc}(\text{Inc_Desc}, \text{" mailbox quota exceed"}) \\ & \rightarrow \text{Resolution}(\text{Resolution_Desc}, \text{" remove mail from mailbox"}) \\ & [\text{Support} = 2\%, \text{Confidence} = 16\%] \end{aligned}$$

代表當情境為使用者部門為“RD 部門”，且事件描述為“信箱爆炸”，常採用的解決方案為“移除使用者的信件”，可信度為 16%，支持度為 2%。

4.2.2.2 Context Relevant Incident Dependency Knowledge Pattern Discovery

為了探勘事件間的關係於某種情境下的關係，我們限制規則的格式，訂定 Meta-Rule 格式如下所示：

$$\begin{aligned} & [\text{Context}(\text{C}, \text{X})]^* \wedge [\text{Staff_Inc}(\text{I}_i, \text{P}_i)] \rightarrow [\text{Staff_Inc}(\text{I}_j, \text{P}_j)] \\ & [\text{Support} = s\%, \text{Confidence} = c\%] \end{aligned} \quad (4.3)$$

根據前面的資料分類，在 Meta-Rule 格式中，規則的左手邊為情境類的資料以及管理者事件分類類別的資料，右手邊則為管理者事件分類類別的資料，其中 **C,I_i, I_j** 代表該類別中的資料屬性名稱，而 **X,P_i,P_j** 則是代表該資料屬性的值。此外，符號 “*” 表示可能包含一個或多個資料屬性，因此在一條規則中，可能會出現一個或多個情境類別中的資料及使用者事件分類類別的資料。如此一來，符合這樣限制的規則將會被找出來，如下所示：

Context(C_Dept, "Accounting") \wedge Staff_Inc (Inc_Desc, " PC Performance Low ")

→ Staff_Inc (Inc_Desc, "PC Crash") [Support = 1%, Confidence = 30%]

表示當情境為“會計部門”，發生的事件描述為“PC 效能低”，常常跟事件描述為“PC 當機”的事件具有相依性，可信度為 30%，支持度為 1%。下表 4.6 為完整的知識樣版例子。

表 4.6 知識樣板範例

Context Relevant Incident Classification Knowledge Pattern	
Example	Interpretation
Context(User_Dept., " DepartmentSale") → Incident(Inc_Desc., "Virus Affected") [support = 2 %, confidence = 30 %]	當情境為銷售部門時，常發出個人電腦中毒的事件。
Context(Service, " ProcurementService") → Incident(Inc_Desc., "Program No Response") [support = 2 %, confidence = 30 %]	當情境為採購服務時，常發生程式執行無回應的事件。
Context(User_Dept, " DepartmentRD") \wedge Context(Service, " E-mail Service") → Incident(Inc_Desc., "Server Performance Low") [support = 2 %, confidence = 30 %]	當情境為研發部門使用 E-mail 服務時，常發出伺服器效能太低的 E-mail 服務事件。
Context(System, " (System_A") \wedge Context(Status, " Server Problem") \wedge Context(Service, " E-mail Service") → Incident(Inc_Desc., "DNS Server Not Work") [support = 2 %, confidence = 30 %]	當情境為使用電子郵件服務，且系統 A 狀態為伺服器有問題時，常發生 DNS 伺服器不運作的事件。
Context(Location, " Taipei") \wedge Context(Service, " AccountingService") → Incident(Inc_Desc., "Unable to Login") [support = 2 %, confidence = 30 %]	當情境為，使用會計服務，且地點在台北的辦公室，常常發生無法登入系統的事件。
Context Relevant Incident Resolution Knowledge Pattern	
Example	Interpretation
Incident(Inc_Desc., " Server Performance Low") \wedge Context (Location, "Taipei") → Resolution(Resolution_desc., "Restart the Server") [support = 2 %, confidence = 30 %]	症狀為伺服器效率太低的事件，當情境狀況為地點在台北辦公室，採取的解決方法是把伺服器重新啟動。
Incident(Inc_Desc., "Unable to send EMail") \wedge Context (C_Dept., "RD") → Resolution(Resolution_desc., "add User' s Mailbox Space") [support = 2 %, confidence = 18 %]	症狀為無法寄信，當情境狀況為 RD 部門員工時，採取的解決方法是把使用者的信箱加大。
Context Relevant Incident Dependency Knowledge Pattern	
Example	Interpretation
Incident(Inc_Desc., "PC Performance Low") \wedge Context (System, "PC_04") → Incident (Inc_Desc., "PC Crash") [support = 2 %, confidence = 28 %]	當情境為系統是"PC_04"，發生事件描述為"電腦效能低"，常常與事件描述為"電腦當機"的事件有相依關係。
Incident(Inc_Desc., "Mailbox Quota exceed") \wedge Context (Service, "EMailService") → Incident (Inc_Desc., "Unable to send email") [support = 2 %, confidence = 28 %]	當情境為使用 Email Service，發生事件描述為信箱爆炸，常常與事件描述為無法寄信的事件具有相依關係。

4.3 推論模組(Inference Module)

推論模組負責協助本架構中推論的部分，如下圖 4.6 所示，在此採用規則式推論方法，包含了規則庫，推論引擎，事實值以及相關的後項(Relevant Consequent)。其中**事實值**來自於服務台得到的事件資訊，情境模組得到的情境資訊，以及情境認知知識支援模組中使用者選擇的知識樣板。**規則庫**則是由資料探勘模組所得的知識樣板以及情境模組中的情境相依關係所組成，**推論引擎**則是負責推論工作，根據輸入的事實值以及規則庫中的規則，推論事實值符合的規則，然後將符合規則的後項根據確定因子值(Certainty Factor Value)高低來排序，得到相關的後項，將這些後項輸出給**相關的後項**，然後送給情境模組以及情境認知的知識支援模組。

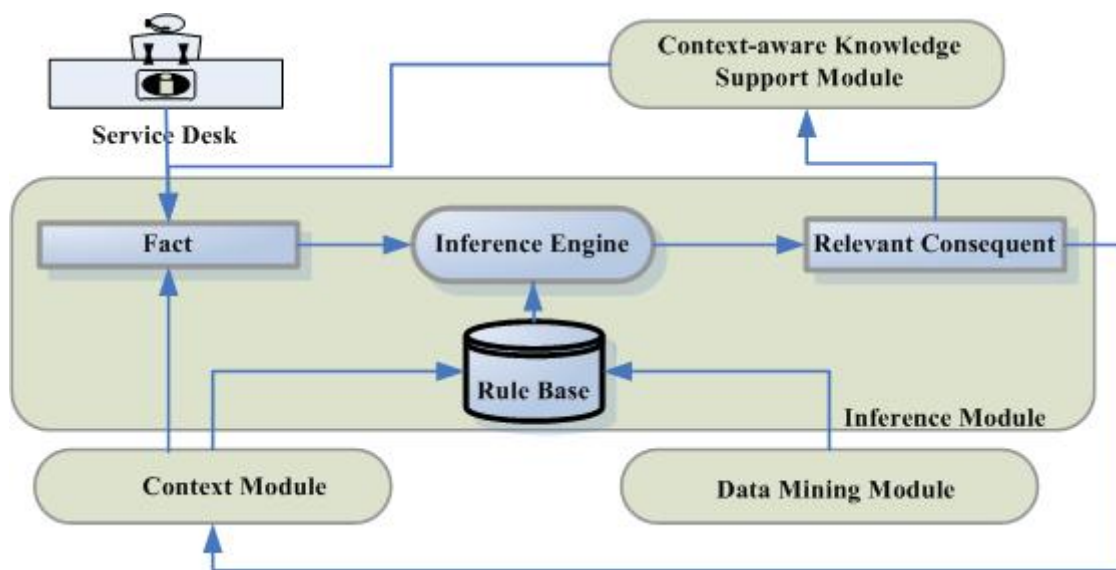


圖 4.6 推論模組架構(Inference Module)

4.3.1 規則式推論(Rule-based Reasoning)

規則式推論方法中，有四個主要的元件，如圖 4.7 所示，包含規則庫、推論引擎、事實值以及相關的後項。下面針對每個元件作詳細的介紹。

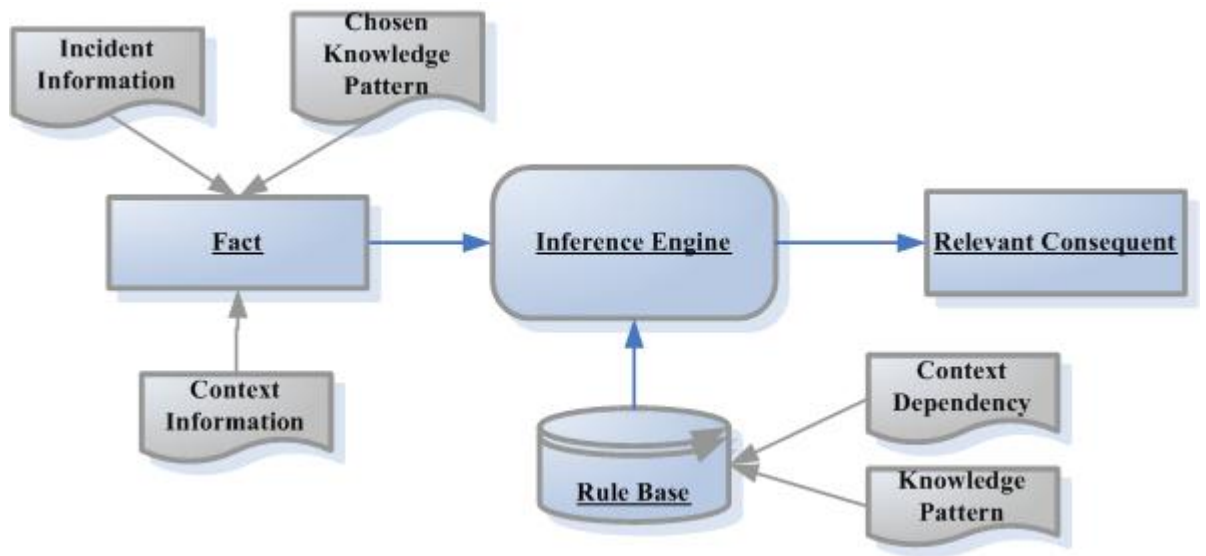


圖 4.7 規則式推論模型(Rule-based Reasoning)

4.3.2事實值(Fact Value)

事實值是推論模組的資料來源，在本架構中，事實值的來源有三個，如上圖 4.7 所示，第一是情境模組的情境資訊，第二是服務台得到的事件資訊，第三是在事件處理過程中，使用者選擇的知識樣板。

舉例來說，Mary 提出使用電子郵件服務時遭遇問題的事件。因此，得到的事實值如表 4.9 所示，包含事件發生時的情境資訊，使用者為“Mary”，職務角色為“Employee”，地點為“MB205”，以及服務為“E-Mail Service”等情境資訊。另外，在事件資訊方面，假設使用者“Mary”提供的使用者描述內容為”Unable to send email”、系統出現的錯誤訊息內容為”Host not found”，以及此項事件的影響程度，輸入值為”High”，急迫程度，輸入值為”Level 2”，及優先程度，輸入值為”1”。最後，當處理事件時，使用者所選取的事實值知識樣版，假設選取的事件描述知識樣版為”Mail Server Shutdown”的事件描述，選取的解決方案描述為”Restart the MailServer”。藉由這些事實值內容，開始進行推論。

4.3.3 規則庫(Rule Base)

規則庫存放推論模組所需的規則，規則庫的來源有兩個，如上圖 4.7 所示，第一是情境模組的**情境相依資訊**，第二是資料探勘模組得到的**知識樣版**。情境相依的規則，如前面 4.1 節所示，描述情境之間的關係。舉例來說，System_A1 由 ServerA 所組成，以規則的方式呈現為“System_A1 → ServerA (Assmble)”，同樣的，規則“Mary→EmailService (Use)”，則是表示使用者 Mary 使用 E-Mail 服務。知識樣版的規則，由資料探勘所得，包含三類知識樣版，如前 4.2 節所示，規則“Mary →Unable to send email, mailbox quota exceed[S=2%, C=15%]”，代表使用者“Mary”常常發生“無法寄信，信箱容量不足”，而規則的可信度為 15%，支持度為 2%。下表 4.8 有完整的規則庫例子。

表 4.7 事實值來源

Context Information		Incident Information		Chosen Knowledge Pattern	
Person	Mary	User_desc	Unable to send email	Inc_desc.	Mail Server Shutdown
Role	Employee	System_Error_Msg	Host not found	Resolution_desc	Restart the MailServer
Dept.	Accounting	Impact	High	Supporter_Name	Clay
Location	MB205	Urgency	Level 2		
Service	E-Mail Service	Priority	1	S_dept.	MIS2
System	System_A1	.	.		

表 4.8 規則庫來源

Context Dependency	Knowledge Pattern
System_A1 → ServerA (Assmble)	C.R. Incident Classification Rule
System_A2 → SwitchB (Assemble)	Mary → Unable to send email, mailbox quota exceed [S=2%, C=15%]
System_A1 → PC005 (Assemble)	RD, EMailService → mailbox quota exceed [S=1%, C=25%]
Mary → EmailService (Use)	MB205, EmailService → MailServerCrash[S=2%, C=18%]
Mary → System_A2 (Use)	C.R. Resolution Rule
PTC → GoldServiceLevel (HasSLA)	Unable to send email, mailbox quota exceed →Extract Mail [S=2%, C=15%]
ServerB → R301 HC (LocatedIn)	mailbox quota exceed, Taipei → remove mail from mailbox [S=1%, C=25%]
	C.R. Incident Dependency Rule
	Unable to send email →MailServerCrash [S=2%, C=15%]
	PC Crash, System_D →CPU Hot [S=2%, C=18%]

4.3.4 推論引擎與相關的後項(Inference Engine and Relevant Consequent)

推論引擎負責將事實值比對規則庫中的規則，得到符合的規則(Matched Rule)，並將符合規則的後項，依據確定因子值(Certainty Factor)高低排序(確定因子簡稱為“CF”)，得到**相關的後項**，再將這些相關後項提供給其他模組。

首先，我們先介紹在推論過程中，需要計算CF值的元素以及他們的表示方法。如表4.9所示，我們計算不確定的證據(Uncertainty evidence)、規則以及推論得到的後項的CF值，而他們的表示法分別為“CF(X)”、“CF(X ⇒ Y)”與“CF(X|R)”。其中，CF值範圍為[-1,1]，數值“1”代表完全確定，數值“-1”代表完全不確定。在不確定證據方面，由於情境資訊都是從環境中所偵測到的，因此我們對於情境資訊的CF值都設為“1”，例如情境資訊，使用者“Mary”的CF值，CF(“Mary”) = 1。

表 4.9 關聯規則中CF表示法

CF (X)	CF of a uncertain evidence
CF(X ⇒ Y)	CF of a rule
CF(X R)	CF of X given the rules in the set R, X is a non-initial fact

我們採用Shortliffe與Buchanan[14]提出的CF值計算方式，計算規則的CF值。計算公式如下所示，S(Y)代表Y的支持度(Support)，Conf(X ⇒ Y)代表規則 (X ⇒ Y) 的可信度(Confidence)。

$$CF(X \Rightarrow Y) = \begin{cases} \frac{(\text{Conf}(X \Rightarrow Y)) - S(Y)}{1 - S(Y)} & , \text{ if } \text{Conf}(X \Rightarrow Y) > S(Y) \\ \frac{(\text{Conf}(X \Rightarrow Y)) - S(Y)}{S(Y)} & , \text{ if } \text{Conf}(X \Rightarrow Y) < S(Y) \\ 0 & , \text{ Otherwise.} \end{cases}$$

舉例來說，計算規則 “Mary → Unable to send email, mailbox quota exceed”的CF值，首先輸入公式中所需的資訊，包含規則的可信度為“15%”，以及不確定證據“Unable to send email, mailbox quota exceed”的支持度為“10%”，代入公式，得到規則“Mary → Unable to send email, mailbox quota exceed”的CF值為“0.05”，計算過程如下。

Mary \Rightarrow Unable to send e-mail [S=2%, C=15%]

S(Unable to send e-mail) = 10%

$$\begin{aligned} \text{CF}(\text{Mary} \Rightarrow \text{Unable to send e-mail}) &= \frac{(\text{Conf}(X \Rightarrow Y)) - S(Y)}{S(Y)} \\ &= \frac{15\% - 10\%}{1 - 10\%} = 0.056 \end{aligned}$$

在推論過程中，我們需要掌握各個元素的 CF 值，才能有效的推論。下面，我們介紹推論過程中，各種 CF 值的計算方法。如下表 4.13 所示，當需要計算兩個交集的不確定證據(X)、(Y)的 CF 值，我們取兩個 CF 值的最小值。如果需要計算兩個聯集的不確定證據(X)、(Y)的 CF 值時，我們取其 CF 值的最大值。如果需要計算後項的 CF 值時，則是將前項 CF 值乘以規則 CF 值得知。當有多個符合的規則具有相同的後項時，我們取其中 CF 最大值當作該後項的 CF 值。

表 4.10 推論過程中，CF 值算法

$\text{CF}(X \wedge Y) = \min(\text{CF}(X), \text{CF}(Y))$	(4.6)
$\text{CF}(X \vee Y) = \max(\text{CF}(X), \text{CF}(Y))$	(4.7)
$\text{CF}(B \{ \text{IF } A \text{ THEN } B \}) = \text{CF}(A \Rightarrow B) \times \max(0, \text{CF}(A))$	(4.8)
$\text{CF}(B) = \max(\text{CF}(B \{ \text{IF } A \text{ THEN } B \}), \text{CF}(B \{ \text{IF } C \text{ THEN } B \}))$	(4.9)

舉例來說，在一個推論過程，如下圖 4.8 所示，我們要計算各個元素的 CF 值，套用上面的公式以及規則，得知結果如圖所示，計算過程如下：

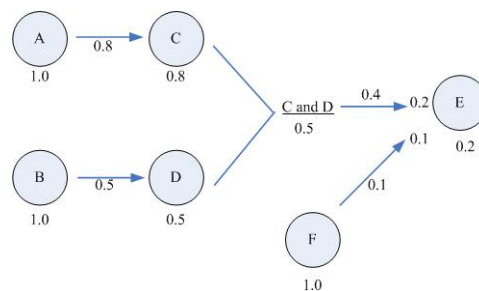


圖 4.8 計算 CF 值例子

$$CF(C) = CF(A) * CF(IF A THEN C) = 1.0 * 0.8 = 0.8$$

$$CF(D) = CF(B) * CF(IF B THEN D) = 1.0 * 0.5 = 0.5$$

$$CF(C \wedge D) = \min(CF(C) , CF(D)) = 0.5$$

$$CF(E) = \max(CF(C \wedge D) * CF(IF C \wedge D THEN E), CF(F) * CF(IF F THEN E))$$

$$= \max(0.5 * 0.4, 1.0 * 0.1) = 0.2$$

根據推論結果，並依照上述 CF 值計算方法，將得到的相關後項，按照 CF 值高低排序，呈現推論的結果。如下表 4.14 所示，在情境相依關係中，由於每個情境的 CF 值都為 1.0，因此符合規則中後項的 CF 值也都為 1.0，相關的後項包含”ServerA”、”PC005”以及”System_A1”。至於知識樣板中，相關的後項包含” MailServer Crash”以及“Unable to send email, mailbox quota exceed”等，依據 CF 值高低排序，表示其相關性的高低。

表 4.11 符合的規則後項，以CF值排序

Context Dependency		Knowledge Pattern	
Consequent	CF Value	C.R. Incident Classification Rule	
ServerA	1.0	Consequent	CF Value
PC005	1.0	Incorrect Username or PWD	0.25
System_A1	1.0	Mail Server Crash	0.17
		C.R. Resolution Rule	
		Consequent	CF Value
		Restart Mail Server	0.16
		Remove Mail From Mailbox	0.12
		C.R. Incident Dependency Rule	
		Consequent	CF Value
		Can't Start Mail Server	0.18

4.4 情境認知事件管理模組(Context-aware Incident Management Module)

情境認知知識支援模組負責依據推論模組得到的相關後項，採取適當的知識支援，如下圖 4.9 所示。根據事件管理需求，我們訂定了三種事件管理所需的知識支援，情境認知事件辨別知識支援、情境認知事件解決方案知識支援，以及情境認知事件相依知識支援。

情境認知事件辨別知識支援，協助服務台人員針對新的事件提供情境認知的事件辨別與事件分類的知識支援。**情境認知事件解決方案知識支援**，協助服務台人員針對新事件提供建議情境認知的解決方案知識支援。**事件感知事件相依知識支援**，協助新事件發生時，提供情境認知的事件相依推論的知識支援，讓服務台人員可以針對多筆相關事件同時處理，或者是對於新事件採取預先處理的動作。

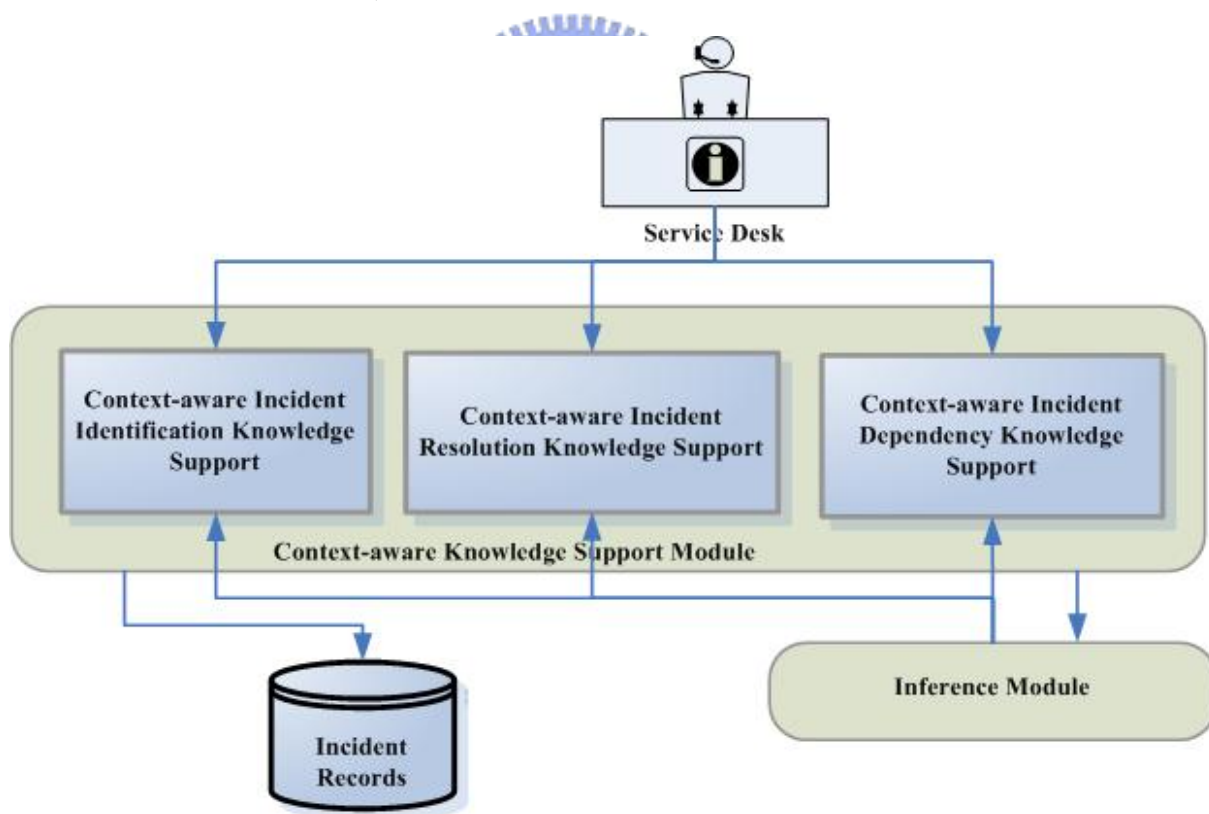


圖 4.9 情境認知知識支援模組架構(Context-aware Knowledge Support Module)

4.4.1 情境認知事件分類知識支援

情境認知事件辨別知識支援，負責協助服務台針對新事件，依據事件發生時的情境以及使用者輸入的事件訊息，經由推論模組推論，推論規則庫中情境相關的事件描述，推薦最有可能的事件描述，供服務台參考。以前面所舉使用者 Mary 發生 E-Mail 事件為例，如下圖 4.10 所示，服務台人員掌握了事件發生時的情境資訊以及使用者提供的事件資訊，透過推論得到可能的事件描述，然後依據 CF 值的高低，推薦最有可能的事件描述，因此，推薦的第一可能事件描述為”Mail Server Shutdown”，CF 值為 0.35，第二可能事件描述為”Wrong User Name or Password”，CF 值為 0.2，將這些事件描述供服務台人員參考。

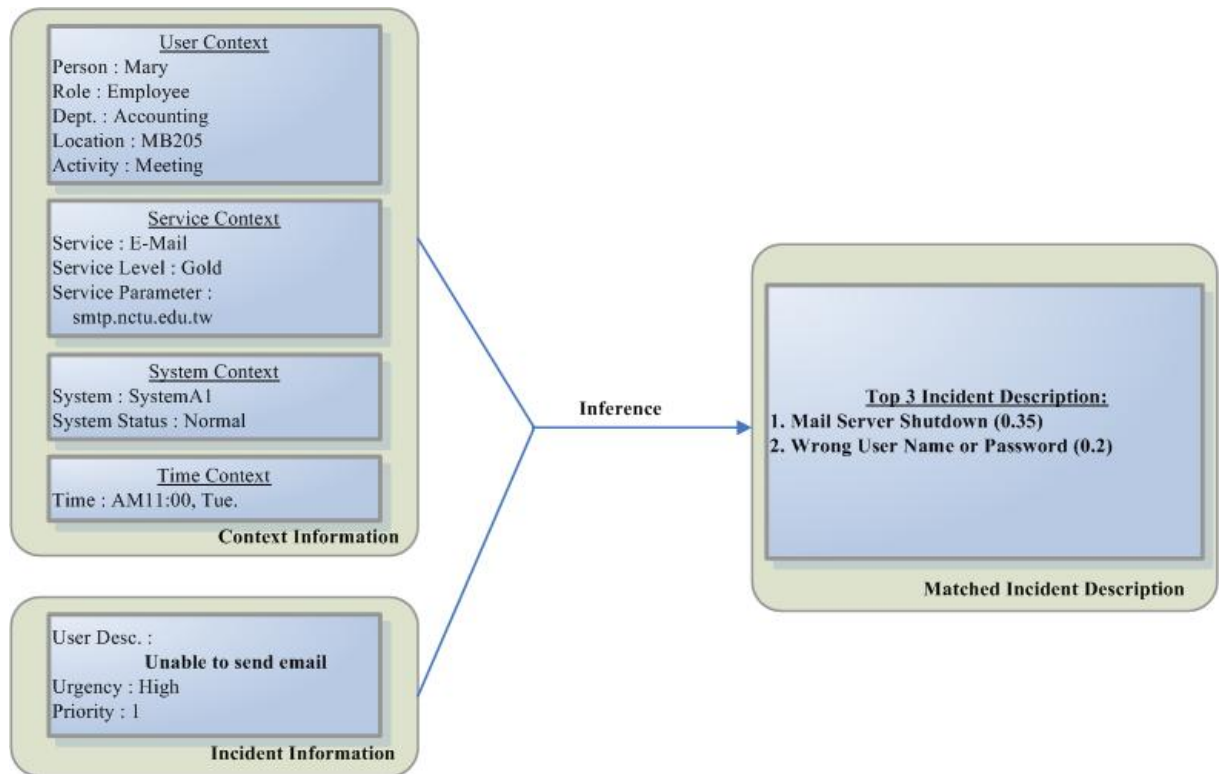


圖 4.10 情境認知事件分類知識支援例子

4.4.2 情境認知事件解決方案知識支援

情境認知事件解決方案知識支援，在服務台針對新事件作完初步的辨別與分類後，提供情境認知的解決方案推薦。同樣的，依據事件的情境資訊、使用者提供的事件資訊，以及服務台針對事件辨別後，所產生的管理者事件描述，然後經由推論模組推論，產出相關的後項，推薦最有可能的解決方案，供服務台人員參考。以前面所舉 Mary 發生 E-Mail 事件為例，如下圖 4.11 所示，由服務台選擇的事件描述，”MailServer Shutdown”，以及事件的情境資訊及事件資訊，經由推論得到的可能解決方案，並依 CF 值的高低，取出最有可能的解決方案，因此，得到的第一可能解決方案描述為”Restart the MailServer”，CF 值為”0.26”，第二可能解決方案描述為” delete useless mail from mail box in mail server”，CF 值為”0.2”，將這些解決方案描述供服務台參考。

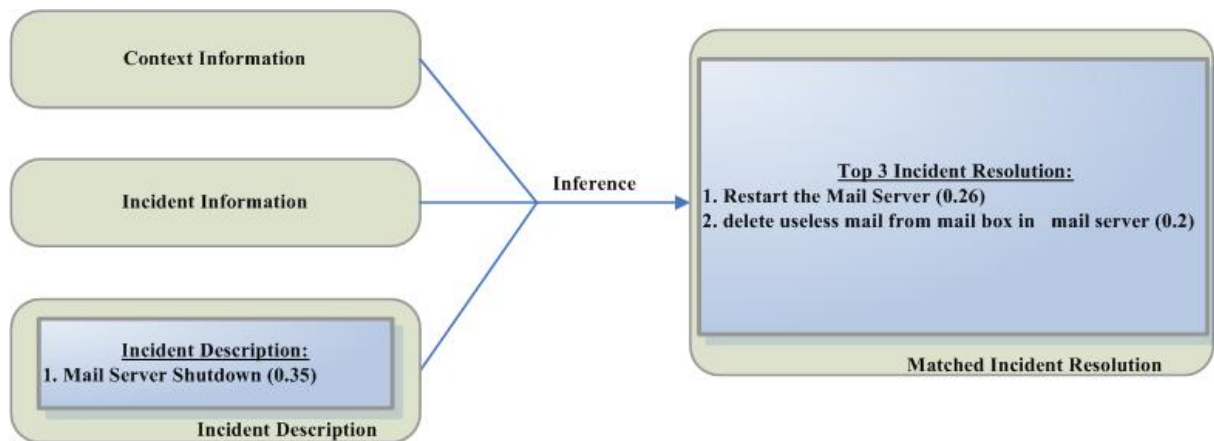


圖 4.11 情境認知事件解決方案知識支援例子

4.4.3 情境認知事件相依關係知識支援

情境認知事件相依知識支援目的在於推薦情境認知的事件相依資訊。當服務台人員針對新事件作完初步的辨別與分類後，除了提供解決方案的建議，情境認知事件相依知識支援同時還提供此事件與其他事件相依性的推薦。藉由這樣的推薦，服務台可以達到兩個目的，第一是將類似事件一併處理，使用一個解決方案，解決多個事件。第二是採取預先處理的解決方案，針對事件未來可能趨勢，預先採取適當的解決方案。

以前面所舉 Mary 發生 E-Mail 事件為例，如下圖 4.12 所示，由服務台選擇的事件描述，"MailServer Shutdown"，以及事件的情境資訊及事件資訊，經由推論得到的可能相依事件，並依 CF 值高低，取出最有可能的相依事件，因此，得到的第一可能相依事件描述為"can't send email"，CF 值為"0.7"，第二可能相依事件描述為" mail server crash"，CF 值為"0.3"，將這些相依事件描述供服務台參考，採取適當的解決方案。

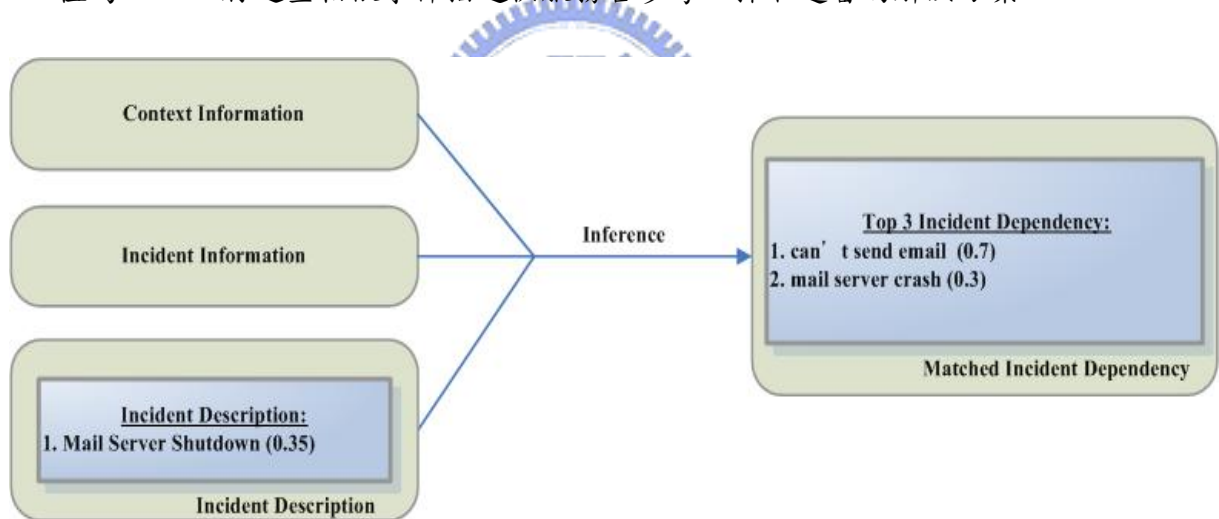


圖 4.12 情境認知事件相依知識支援例子

五、系統建置與實作

依據前面所提的架構，我們設計一個雛型系統，來說明整個架構的運作方式。本章中詳述系統實作方式、開發工具及平台與系統功能介紹，並且使用Web介面的情境認知服務台，展示事件處理時的情境認知知識支援。最後以一個模擬的事件，說明本系統如何對於事件處理進行有效的協助。

本章節將分成五個部份，第一個部分將說明系統的實作方式以及開發工具與平台。第二部份是情境模組的實作說明，說明如何使用Protégé建置情境本體類別(Context Ontology class)。第三部分則是資料探勘模組的實作說明，說明如何使用Weka進行有限的關聯規則探勘，找出對事件管理有用的情境相關知識。第四部份則是以一個模擬的事件，以Web介面呈現的服務台，說明推論模組的運作方式，以及整個情境認知的事件處理知識支援。

5.1 開發工具及平台



本雛型系統的開發工具與平台如下表5.1所示：

表 5.1：系統開發工具與平台

系統平台	Operating System	Microsoft Windows Server 2000
	Application Server	Tomcat 4.0
	Database Server	Microsoft SQL Server 2000
開發語言與工具	JSP 2.0 Java JDK 1.3	
情境資訊呈現語言	Ontology Web Language(OWL)	
資料探勘工具	Weka 3-4	
情境資訊建置工具	Protege 3.1	

5.2 情境模組的實作內容

在此說明如何使用Protégé建置情境本體類別(Context Ontology Class)以及情境相依關係，並且依據這些類別，產生我們所需的案例(Instance)，然後以OWL語言呈現，達到共享(Reuse)以及機器可讀(Machine Readable)等優點。最後以RD部門的Mary使用者在使用E-mail服務發出事件請求為例，說明此事件的情境資訊，以及以OWL呈現的結果。

我們使用Protégé 建置情境的本體類別，Protégé[21]是由史丹佛(Stanford)大學所開發，是一個以Java為基礎，開放原始碼(Open Source)的本體編輯器(Ontology Editor)，其中的OWL外掛程式(Plugin)[22]，可以讓我們使用Protégé來讀取、儲存與編輯OWL本體類別。OWL使用於應用程式之間的訊息傳遞，與其他語言比較，如XML、RDF等，OWL提供更多額外的詞彙於正式的語意間，來描述屬性以及類別。

下面我們介紹以Protégé進行情境造型(context modeling)，依據前面4.1節所設計的情境特徵本體與情境相依關係，產生情境本體類別：

(1). 新增情境本體類別：

在這裡我們依據先前4.1節所訂定的情境特徵本體，設計本體類別，如圖5.1所示，設計了下面幾個類別，Location、Organization、Person、Service、SysComponent、System以及Time類別。

(2). 設計情境特徵類別的資料屬性

設計完情境特徵的本體之後，開始填入每個情境特徵本體的資料屬性(data properties)，以人員(Person)本體類別為例，資料屬性包含姓名(Name)、性別(Gender)、地址(Address)以及職務角色(UserRole)。如下圖 5.2 所示，我們於Person的類別中，加入上述的屬性(Properties)。

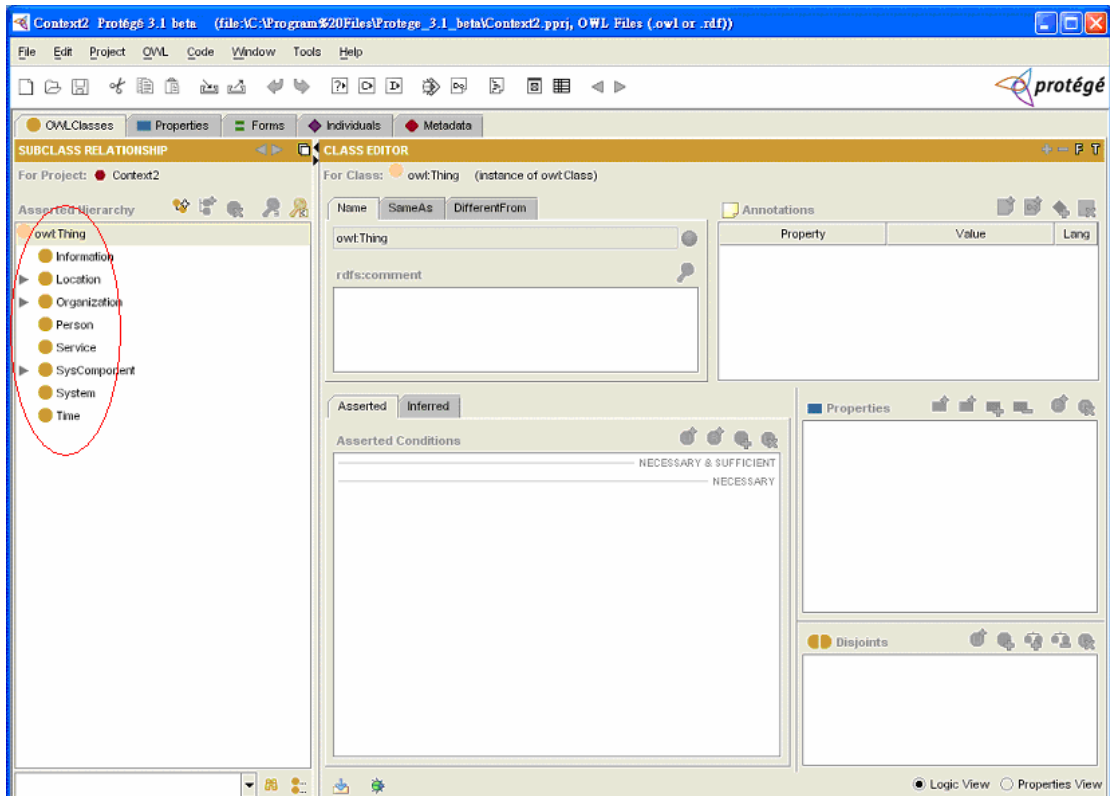


圖 5.1：設計情境特徵類別

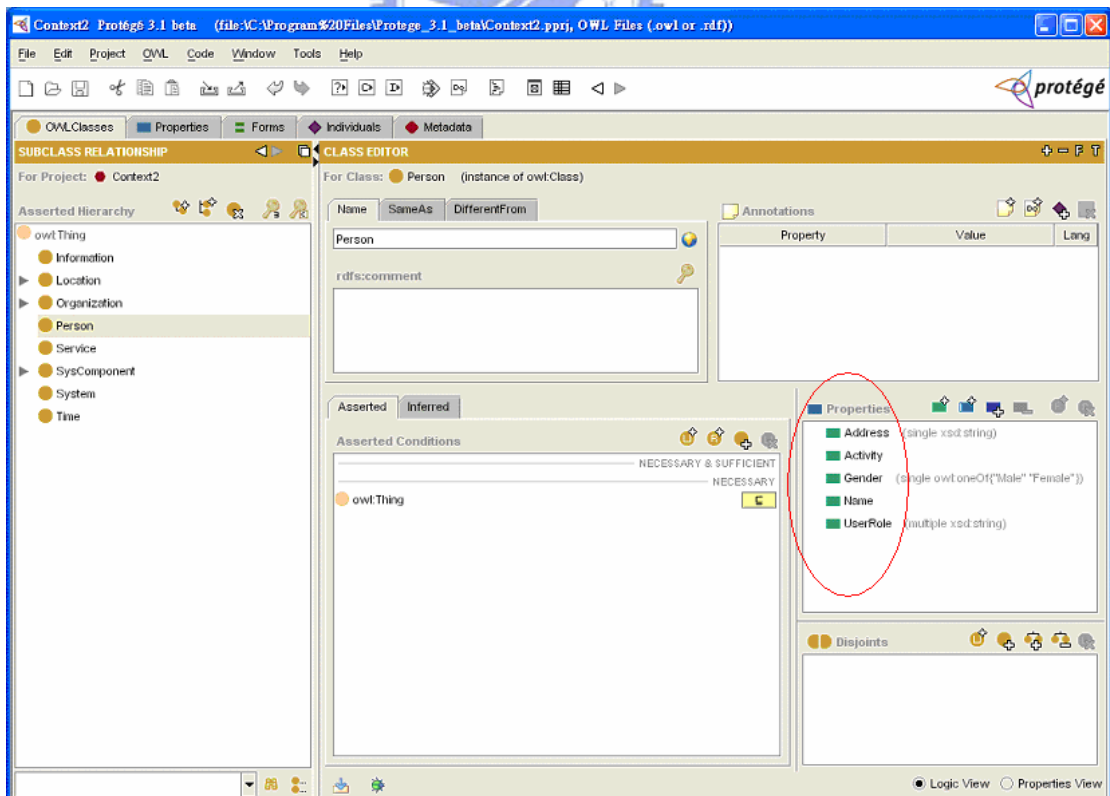


圖 5.2：設計情境特徵類別 Person 的資料屬性

(3). 設計子類別

部分情境特徵類別之間，具備有子類別的關係，在此以SysComponent為例，SysComponent類別包含NetworkDevice、Platform等子類別，如下圖5.3所示，我們使用Protégé設計SysComponent子類別。

(4). 設計情境相依關係

當設計完各個情境特徵類別之後，我們開始設計情境相依關係，舉例來說，類別Person與Service之間，具備情境相依關係”Uses”，因此我們於Properties新增物件屬性(Object Properties)，命名為”UseService”，並設定Domain與Range分別為”Person”與”Service”，表示情境相依關係，“Person → Service (Use)”的關係。如下圖5.4所示。

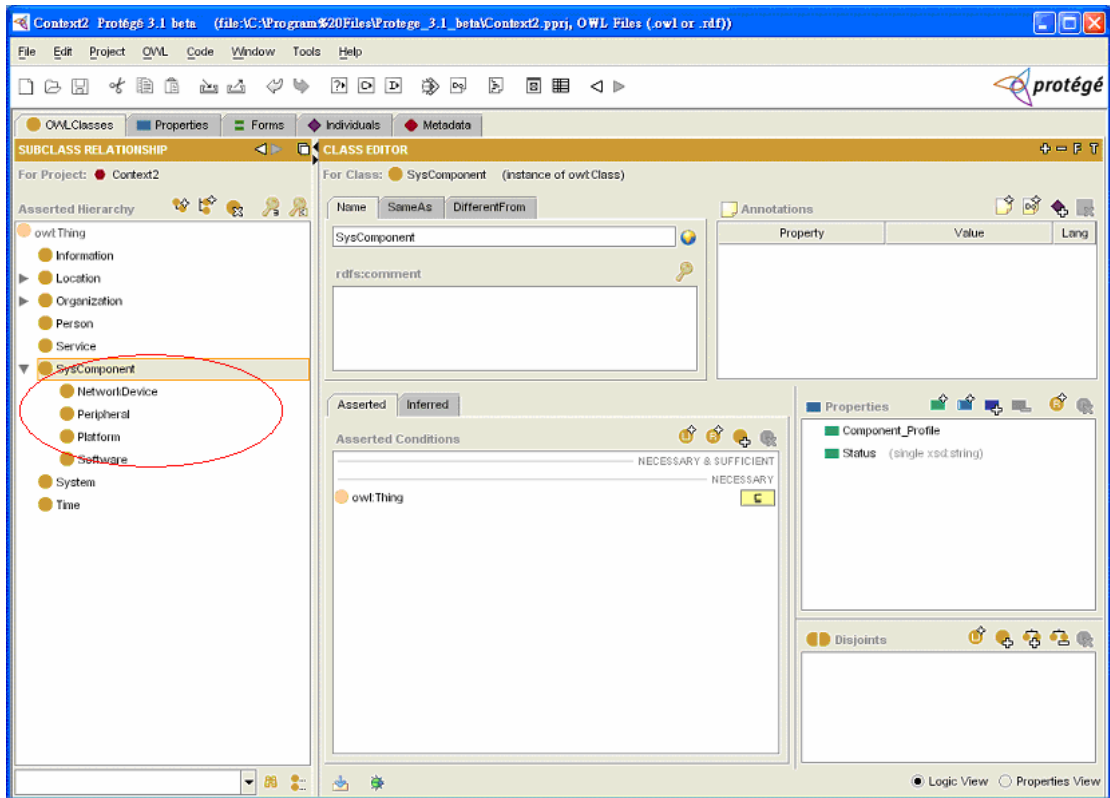


圖 5.3：設計情境特徵類別 SysComponent 的子類別

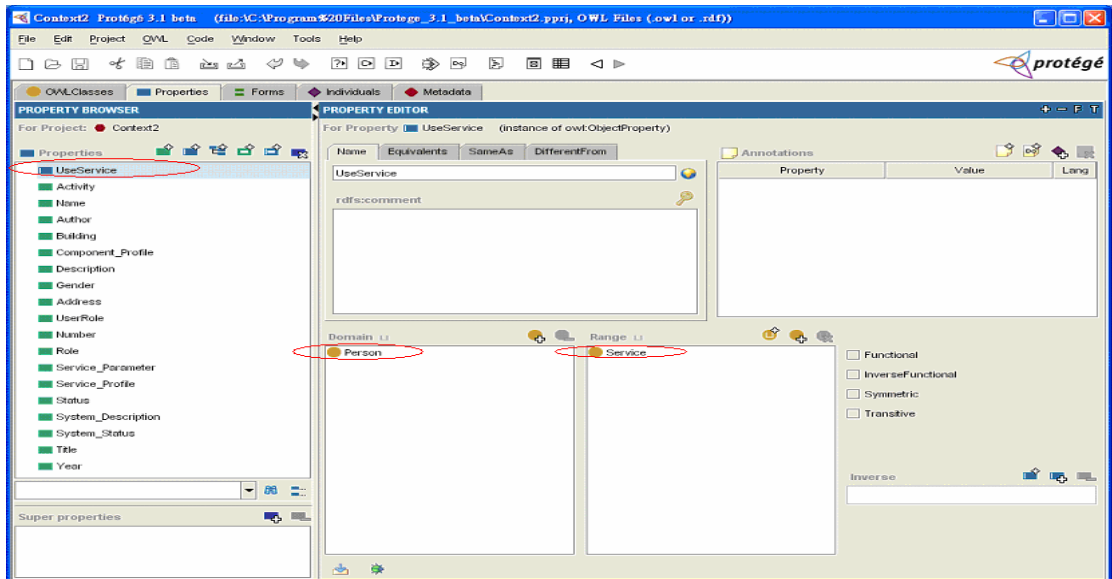


圖 5.4：設計情境相依關係 UseService

(5). 填入各個類別的個體(Individual)

當建立好各個情境特徵類別與情境相依關係之後，開始填入各個類別中的個體，以類別Person為例，我們填入個體為”Mary”，以及他的各項情境資訊以及與其他情境個體的情境相依關係。如圖5.5所示，Mary的屬性資訊Address為”TR No 1001, HC, Taiwan”，Gender為”Female”等，情境相依關係Use_Service關聯的個體為”Email”與”Internet”。

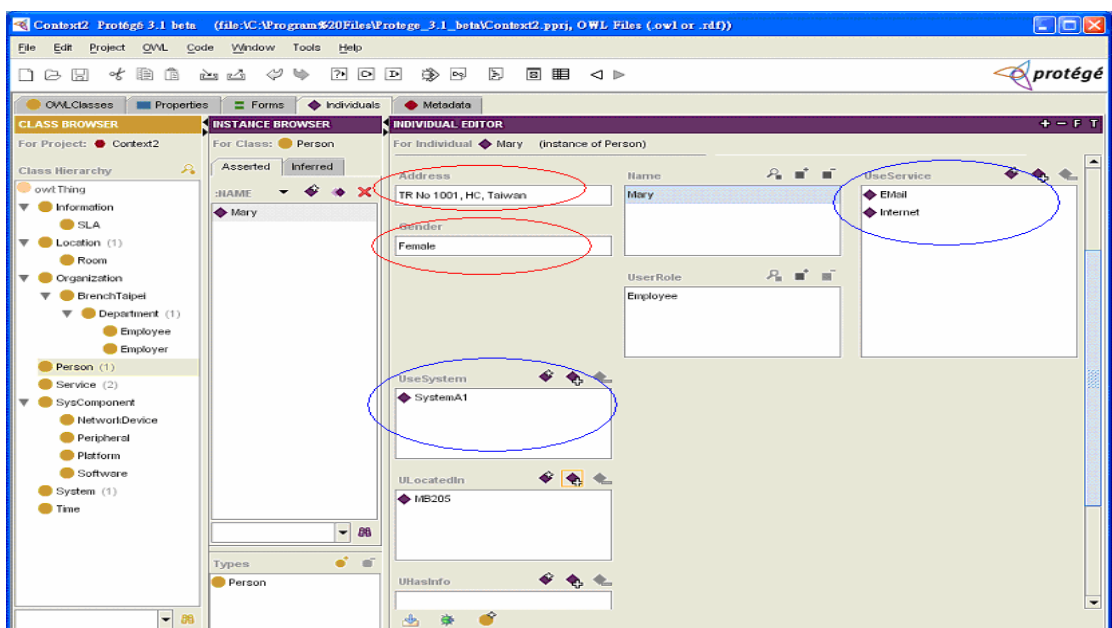


圖 5.5：設計類別個體”Mary”與其他情境資訊

(6). 以OWL展示情境特徵類別及類別個體

以Protégé產生的情境特徵類別、情境相依，以及根據這些類別所產生的個體，其對應的OWL文件，如下圖5.6、5.7、5.8、5.9所示。圖5.6表示SysComponent類別與子類別Network Device的OWL表示方法。圖5.7表示類別Person與類別Location類別之間的情境相依關係為LocatedIn。圖5.8表示類別Person具有屬性資訊Role。圖5.9則是表示個體Mary的情境資訊，包含Mary的屬性資訊Role為Employee，以及情境相依資訊UseService等。

```
<owl:Class rdf:ID="Platform">
  <rdfs:subClassOf rdf:resource="#SysComponent"/>
</owl:Class>
<owl:Class rdf:ID="Employer">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Department"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Information"/>
<owl:Class rdf:ID="System"/>
<owl:Class rdf:ID="Service"/>
<owl:Class rdf:about="#Department">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="BranchTaipei"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="NetworkDevice">
  <rdfs:subClassOf rdf:resource="#SysComponent"/>
</owl:Class>
<owl:Class rdf:ID="Organization"/>
<owl:Class rdf:about="#BranchTaipei">
  <rdfs:subClassOf rdf:resource="#Organization"/>
</owl:Class>
```

圖 5.6：以 OWL 展示子類別關係，以 SysComponent 為例

```
<owl:ObjectProperty rdf:ID="ULocatedIn">
  <rdfs:range rdf:resource="#Location"/>
  <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="UseSystem">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#System"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="BelongTo">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Department"/>
</owl:ObjectProperty>
```

圖 5.7：以 OWL 展示物件屬性，以 ULocatedIn 為例

```
<owl:DatatypeProperty rdf:ID="Role">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="System_Description">
  <rdfs:domain rdf:resource="#System"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="UserRole">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

圖 5.8：以 OWL 展示資料屬性，以 Role 為例

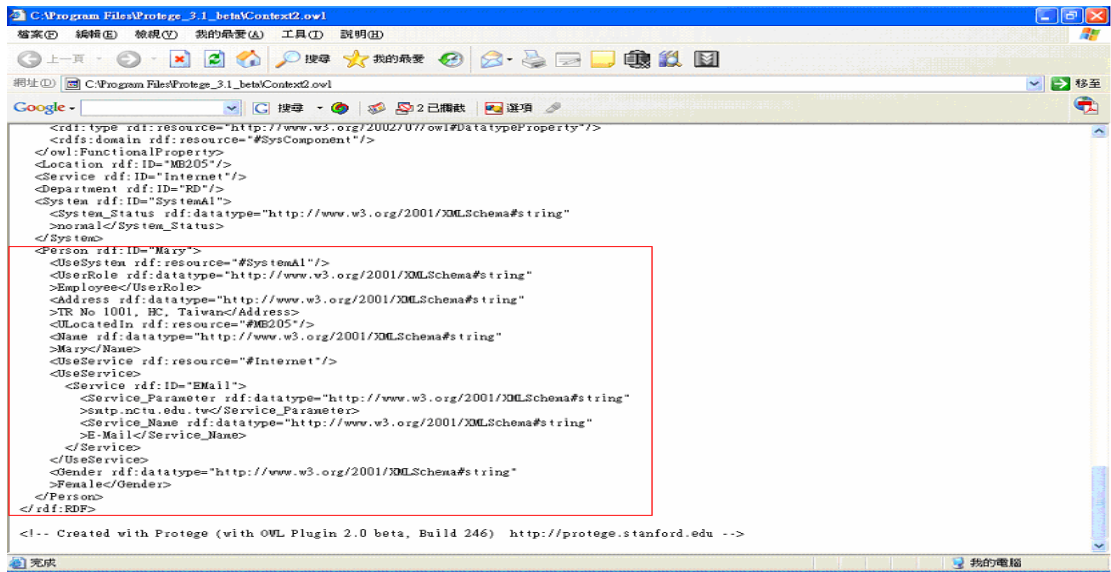


圖 5.9：以 OWL 展示類別個體，以 Mary 為例



5.3 資料探勘模組的實作內容

在此根據前面4.2節所提出的方法以及訂定的Meta Rule格式，對於蒐集到的歷史事件紀錄，以Weka進行關聯規則的探勘，找出我們需要的知識樣板。Weka是一個開放原始碼的資料探勘軟體，以Java為基礎，提供多項資料探勘分析方法供使用者使用，我們使用Weka進行資料探勘，從得到的關聯規則中，找出符合Meta-Rule格式的關聯規則，當作我們的知識樣版。

(1). 整理事件報告資料來源

根據前面4.2小節所示，針對歷史事件報告，挑選我們需要的資料屬性，進行關聯規則探勘。我們所使用的資料庫系統為SQL Server 2000，而資料來源的資料關聯圖如下圖5.10所示。

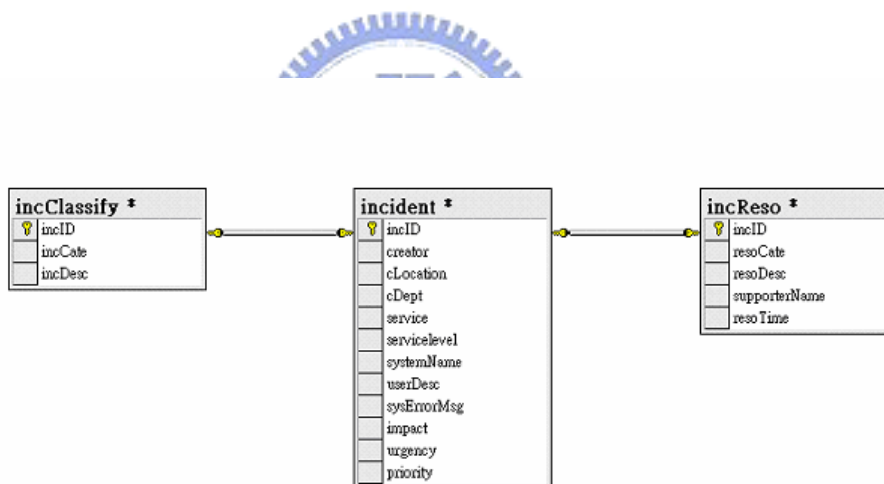


圖 5.10：事件報告的資料關聯圖

(2). 使用 Weka 進行關聯規則探勘

將前面整理好的事件報告，匯入Weka中，圖5.11顯示Weka匯入資訊的畫面，然後開始進行關聯式規則探勘，得到的結果如圖5.12所示。由Weka運算的結果，我們僅挑選符合前面4.2節所訂定的Meta Rule格式的有用規則，當成我們有用的知識樣版。

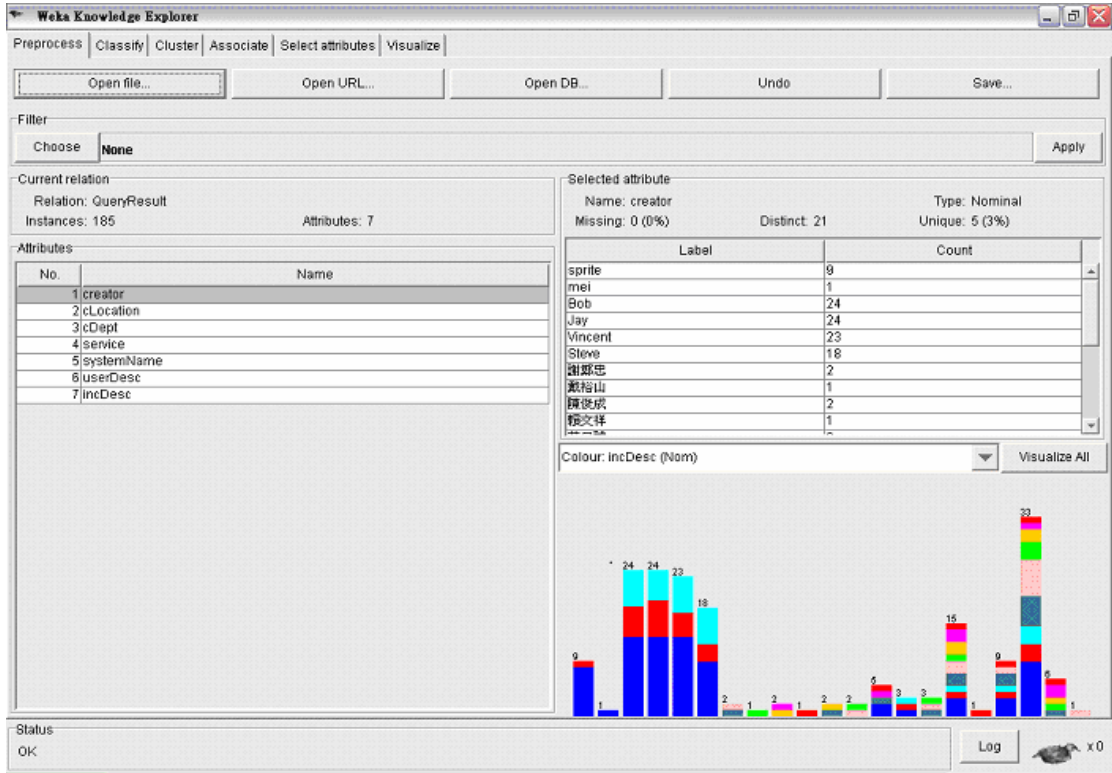


圖 5.11：Weka 讀入事件報告的表示畫面

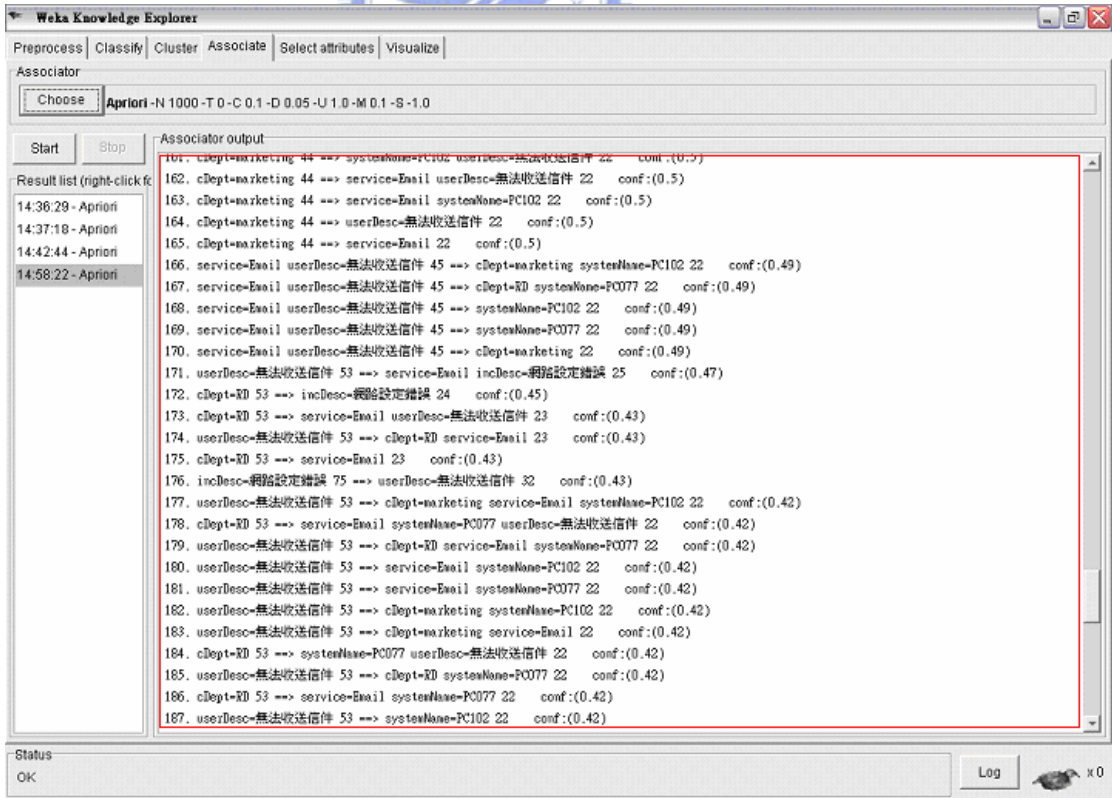


圖 5.12：情境相關事件分類的關聯規則分析結果

5.3 推論模組與情境認知的事件知識支援模組的實作說明

當得到建立的情境模型，以及探勘得到的情境相關知識樣板結果，我們運用JSP[23]來開發Web介面的服務台系統，稱為情境認知服務台(Context-aware Service Desk)。當使用者登入系統時，系統展示使用者目前完整的情境資訊，當使用者提出事件申請單時，可以依據情境資訊，快速且正確的填入事件詳細資訊。另外，當服務台人員處理新事件時，系統依據事件的詳細事件資訊，包含完整的情境資訊，推論得到適當的事件管理知識，協助服務台人員快速解決問題。下面，我們以使用者sprite提出事件申請為例，說明系統如何提供正確且快速的事件記錄協助，以及系統如何根據推論結果，提供服務台人員情境認知的事件分類協助以及事件解決方案協助。

圖5.13為情境認知服務台的首頁畫面，使用者“Sprite”填入正確的帳號及密碼，然後進入服務台的主畫面，如圖5.14所示，主畫面中除了顯示系統的主要功能清單外，在系統的右手邊，顯示使用者此時的情境資訊，包含使用者情境、服務情境、系統情境、以及時間情境。另外，主畫面的中間則是顯示使用者最近提出的事件申請單，供使用者瀏覽。

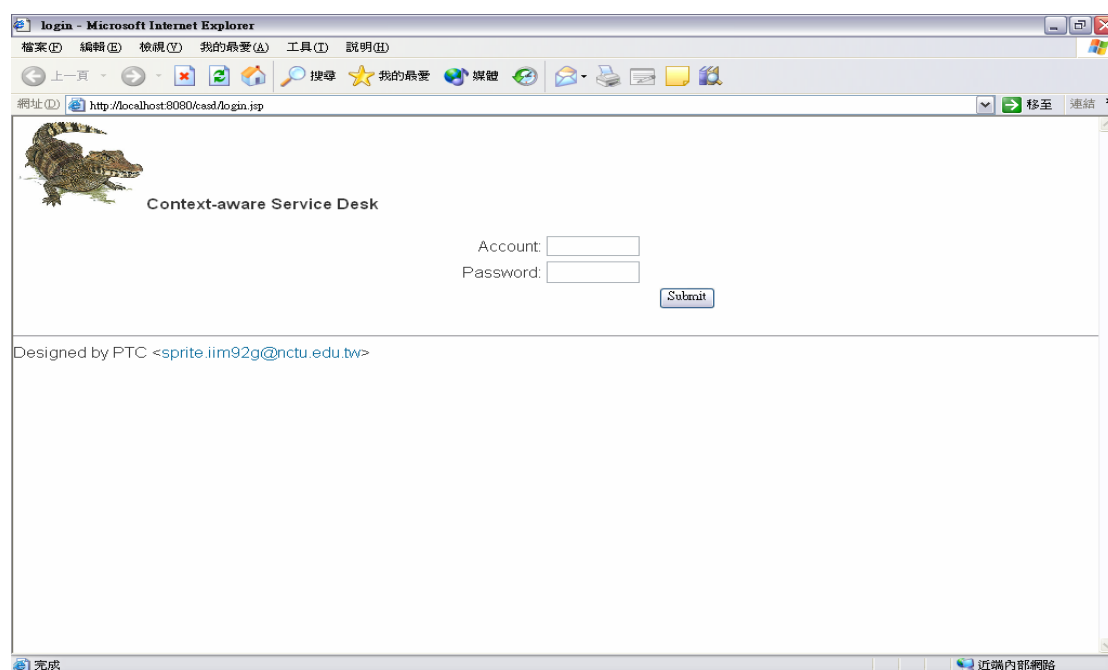


圖 5.13：系統登入畫面

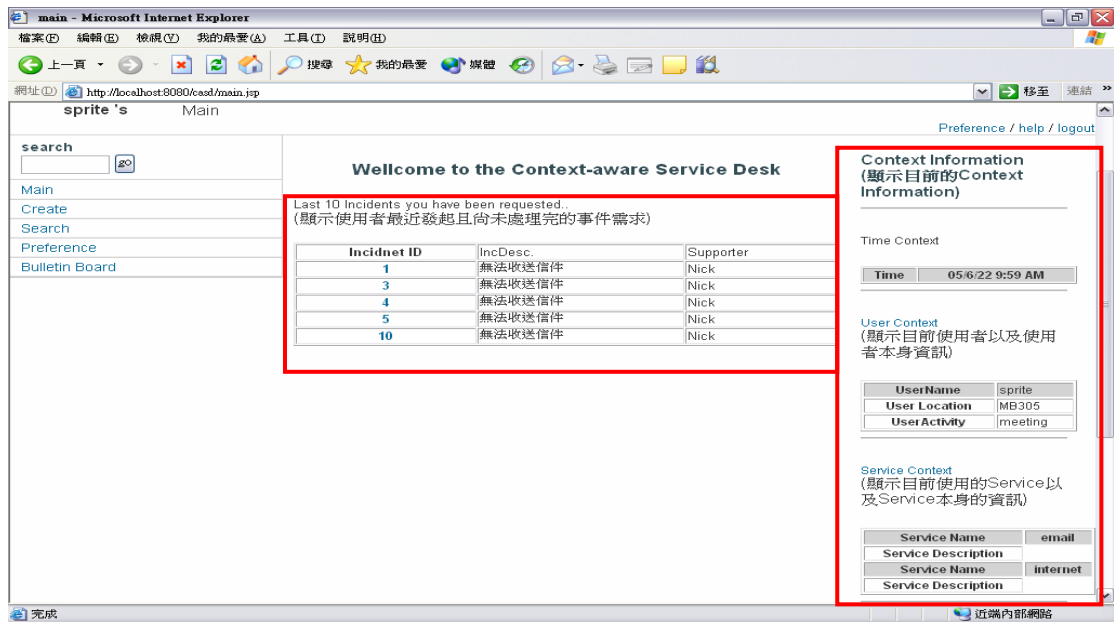


圖 5.14：使用者主畫面，顯示使用者的情境資訊

當我們想要了解更詳細的情境特徵資訊以及情境相依資訊時，我們可以點選的情境資訊中各個情境個體來觀看。舉例來說，當我們點選其中的User Context時，如圖5.15所示，可以得知詳細的使用者情境，使用者“Sprite”的詳細情境特徵資訊包含性別為“Male”、部門為“RD”等，以及與其他情境個體的關係，包含與服務“e-mail”為“使用”的關係等。藉此更了解使用者當時的情境。

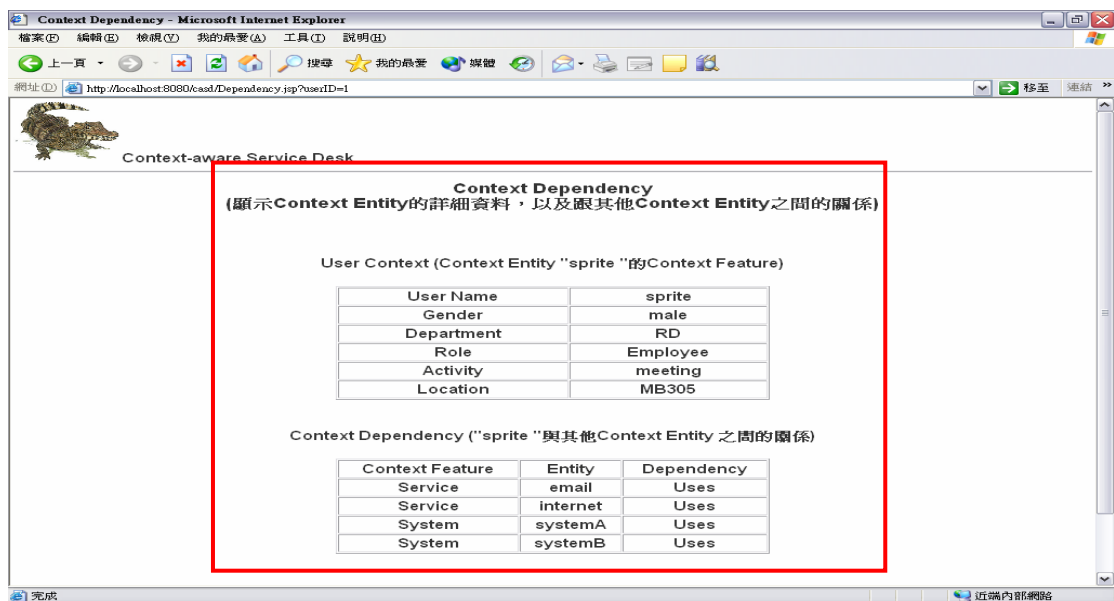


圖 5.15：使用者“sprite”的情境特徵以及情境相依關係

當使用者遭遇事件，需要發起事件需求時，使用者可以點選主頁中右手邊的功能選項“create”，開始填寫事件申請表格。如圖5.16所示，由於系統紀錄使用者詳細的情境資訊，因此，當使用者填寫事件申請表時，系統會先自動填入部分的事件資訊，包含申請者(Creator)、使用者部門(cDepartment)、時間(Time)、地點(Location)，至於其他的事件資訊，系統會依據使用者當時的情境資訊，過濾掉無關的選項，僅提供有關的選項供使用者選取。因此，在本例子中，使用者填寫事件申請表時，系統已經填入部分的事件資訊，而使用者需要填寫的為受影響的系統，提供的選項僅有“PC001”及“SystemA”，受影響的服務，提供的選項則有“E-Mail”及“Internet”供使用者選取。然後使用者再填入對於事件的描述、系統的錯誤訊息、影響程度與緊急程度。填寫完之後的事件詳細資訊如圖5.17所示。

Context-aware Service Desk

sprite Create

Preference / help / logout

search

Main

Create

Search

Preference

Bulletin Board

(填入事件的詳細資訊，系統會依據情境資訊，自動填入適當資訊，以及提供正確的選項，供使用者選取。)

Service: email

SystemName: systemA

UserDesc:

SysErrorMsg:

Impact: High

Urgency: Immediate

Submit

Context Information

Time Context

Time 05/6/13 4:57 PM

User Context

UserName	sprite
User Location	MB305
User Activity	meeting

Service Context

Service Name	email
Service Description	
Service Name	internet
Service Description	

System Context

SystemName	systemA
------------	---------

圖 5.16：使用者填寫事件申請表格

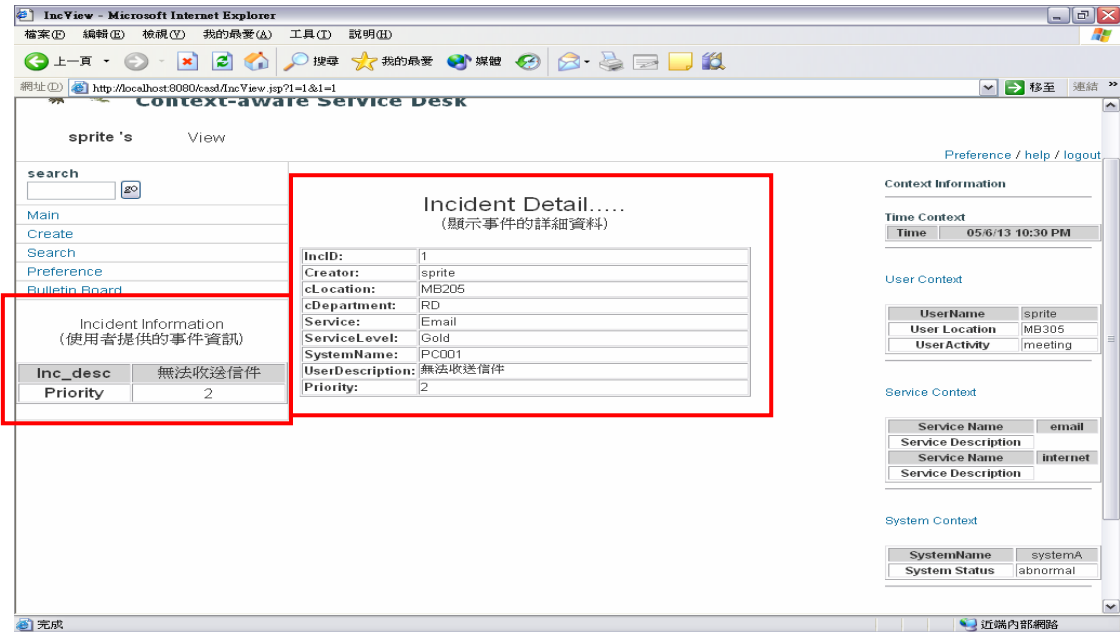


圖 5.17：顯示事件的詳細資料

當服務台人員登入系統時，如圖5.18所示，除了顯示情境資訊以及最近提出的事件申請外，系統下方顯示服務台人員待處理的事件，服務台人員點選待處理的事件單，開始進行事件處理。我們以事件編號1的事件單為例。

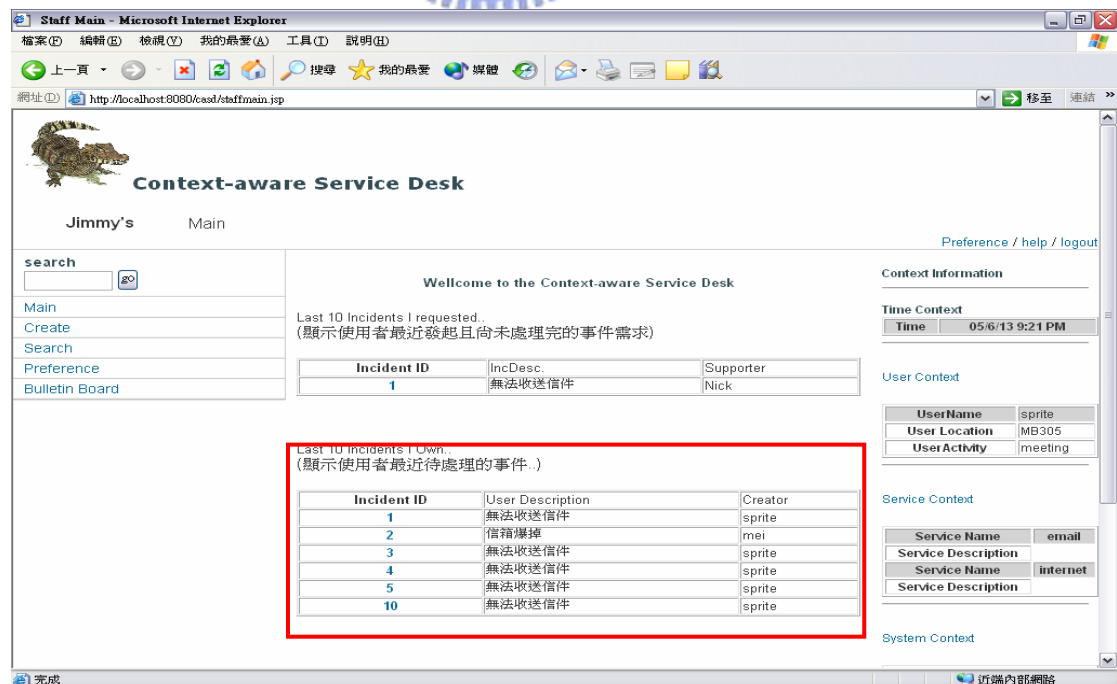


圖 5.18：服務台人員的主畫面

如圖5.19所示，系統除了呈現事件的詳細資訊，以及事件的情境資訊外。系統還依據推論結果，推薦事件最有可能的管理者事件分類描述，供服務台人員參考，了解事件的可能問題原因。在本例子中，系統推薦可能的事件分類則為“信箱爆炸”，CF值為“0.35”，“網路設定錯誤”，CF值為“0.14”，以及“Mail Server故障”，CF值為“0.12”。服務台人員根據這些推薦，檢視事件的詳細資訊，選取正確的管理者事件分類描述。在此，我們假設它選取了推薦的第一個選項，判斷為“信箱爆炸”的事件，然後繼續進行解決事件的工作。

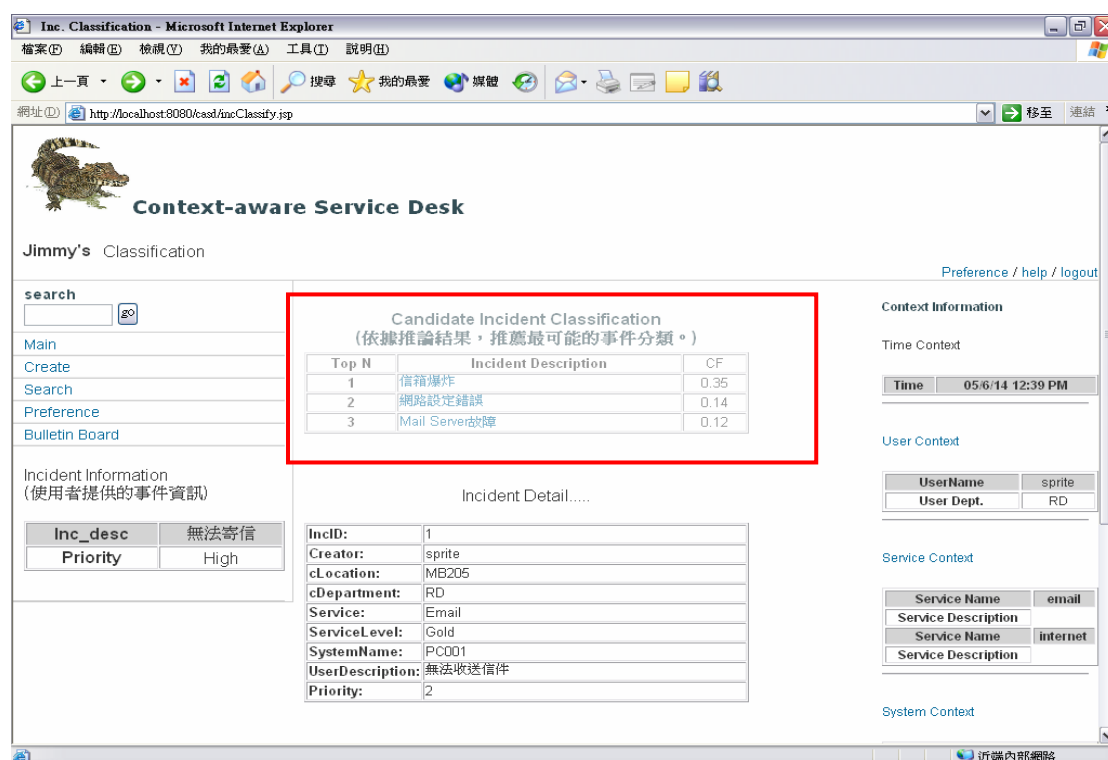


圖 5.19：情境認知的事件分類推薦

當服務台人員判斷事件的管理者事件分類描述之後，他要採取適當的解決方法，將事件解決。此時，系統會依據推論結果，推薦事件最有可能的解決方案，如下圖5.20所示，依據推論結果，系統推薦的解決方案分別為“加大使用者信箱”，CF值為0.165，“移除使用者於Mail Server的信件”，CF值為0.049，以及“壓縮郵件”，CF值為0.044。服務台人員根據這些推薦，檢視詳細的事件資訊，選取有效的解決方案，在此，我們假設它選取了推薦的第一個選項，“加大使用者的信箱”去解決此項事件，也順利的解決事件。

當事情解決後，系統會展示事件的完整訊息，包含從事件記錄開始、事件分類與事件解決等資訊，然後將事件結案，如圖5.21所示。

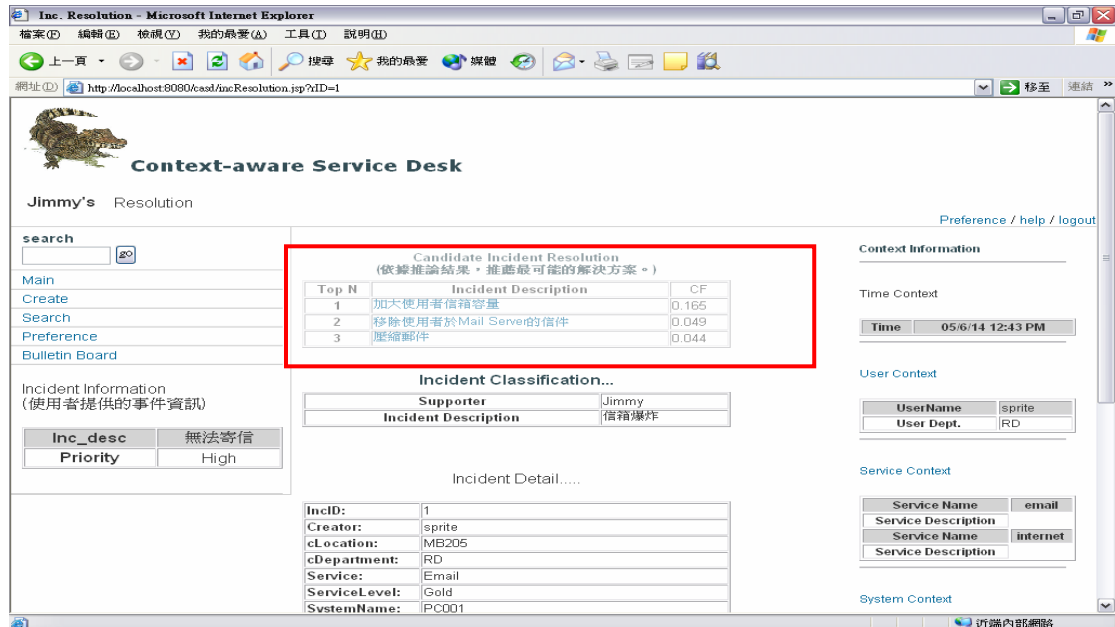


圖 5.20：情境認知的事件解決方案推薦

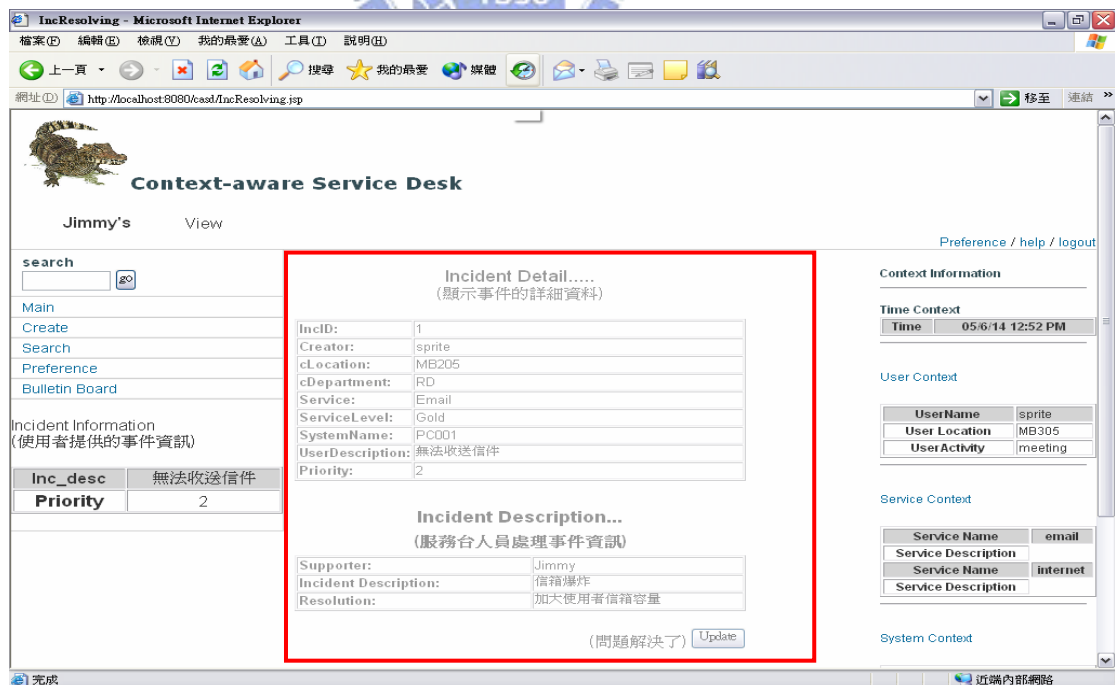


圖 5.21：問題解決畫面

六、結論與未來研究方向

6.1 結論

本研究呈現了一個情境認知的知識支援架構，用於IT服務管理中的事件管理，協助企業進行事件管理活動時，提供適當的知識支援。在提出的架構中，我們使用了OWL語言表示法，來定義IT服務中有用的情境資訊，並使用資料探勘中的關聯式規則探勘技術，探勘事件發生時的情境資訊與事件處理資訊之間的知識樣板。然後使用推論的方法，提供情境認知的事件管理知識支援，協助企業快速的解決新事件的處理。此外，我們依據所提出的架構，建置了一個Web介面的雛型系統，以模擬的案例，說明當新事件發生時，系統如何以豐富的情境資訊，協助使用者快速的紀錄事件資訊，以及當服務台人員處理事件時，系統如何提供情境認知的知識支援，讓服務台人員快速的解決新事件。

本研究的主要貢獻有下列幾點：

- 提出情境認知的知識支援架構，應用於IT服務管理的事件管理中，不僅以ITIL的精神為出發點，更進一步利用情境認知的概念，提供有效的事件管理知識支援，產生更好的事件管理。
- 建構IT服務環境中的情境模型，藉由擷取事件的情境，協助企業在處理新事件時，快速掌握事件相關資訊，增加事件處理的效率，提升IT服務的品質。
- 使用關聯式規則探勘的技術，從事件歷史紀錄中，挖掘出企業內關於事件情境資訊與事件處理資訊間的知識，協助企業從歷史資料中萃取出有用的知識，達到知識複用的目的。
- 建置Web介面瀏覽方式的雛型系統，展示在情境認知的知識支援架構下，系統如何提供情境認知的事件管理知識支援，增加服務台人員事件處理的效率。

6.2 未來研究方向

- 以真實資料測試所設計的架構與系統

本研究的系統礙於完整的真實資料取得不易，因此實作係用假設的資料來模擬情境認知的事件管理知識支援流程。但畢竟假設資料與真實情況仍是會有些差異，因此將來若能與相關企業合作，讓企業提供實際資料與環境來運作，將更能檢驗本論文所提出的情境認知的知識支援架構之可行性，相關的一些研究成果也會更具說服力。

- 評估方法的設計

由於其他限制因素，在本研究中，針對評估部份，尚未做深入的研究，未來可以針對評估方法，進行深入的研究，設計適當的評估方法，讓本研究更加完整。

- 將情境認知的概念，運用於 ITIL 中的其他管理流程

本研究目前僅將情境認知的概念，運用於 ITIL 的事件管理流程。然而，企業在提供完整的 IT 服務管理時，仍有其他管理流程需要注意，例如問題管理流程、變更管理流程等，而在這些管理流程中，情境資訊也能提供良好的協助。因此，未來研究可以將情境認知的概念，運用於其他的管理流程，可能會產生有效的助益。

- 與案例式推論方法(Case-Based Reasoning)整合

相關的研究中，部分學者使用 CBR 的方法來協助進行事件處理或問題管理。本研究中，並未使用 CBR 的方法，未來可以將本研究與 CBR 方法整合，用以建置更有效的知識支援架構。

參考文獻

- [1] R. Agrawal, T. Imielinski, A. Swami, "Mining Association Rules between Sets of Items in Large Databases," *Proceeding of ACM SIGMOD International Conference on Management of Data*, May 1993, pp.207-216.
- [2] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *Research Report RJ 9839*, IBM Almaden Research Center, 1994.
- [3] P. Brézillon, "Context in Problem Solving: A Survey," *The Knowledge Engineering Review* (1999), 14:47-80
- [4] P. Brézillon, C. Gentile, "SART: A system for supporting operators with contextually knowledge," *Proceedings of the First International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)*. Federal University of Rio de Janeiro Ed., 1997, pp. 209-222.
- [5] G. Chen, D. Kotz, "A Survey of Context-Aware Mobile Computing Research," *tech. report TR2000-381*, Dept. of Computer Science, Dartmouth College, Hanover, N.H., 2000.
- [6] A.K. Dey, G.D. Abowd, "Towards A Better Understanding of Context and Context-awareness," *GVU Technical Report GITGVU-99-22*, 1999.
- [7] A.K. Dey, "Understanding and Using Context," *Journal of Personal and Ubiquitous Computing*, vol. 5, no. 1, Feb. 2001, pp. 4-7.
- [8] C. Guanling, K. David, "A Survey of Context-Aware Mobile Computing Research," *Dartmouth Computer Science Technical Report TR2000-381*, 2000.
- [9] T. Gu, HK Pung, DQ Zhang, "A Service-Oriented Middleware for Building Context-Aware Services," *Journal of Network and Computer Applications* 28 (2005) 1-18
- [10] T. R. Gruber, "Towards Principles for the Design of Ontologies Used for Knowledge

- Sharing,” *International Journal of Human-Computer Studies*, Vol.43, No.5-6, 1995, pp.907-928
- [11] M. Khedr, A. Karmouch, “ACAI: agent-based context-aware infrastructure for spontaneous applications,” *Journal of Network and Computer Applications* 28 (2005) 19–44
- [12] N. Ryan, “Mobile Computing in a Fieldwork Environment: Metadata Elements,” *Project working document*, version 0.2 (1997)
- [13] Schilit, Bill and Theimer, Marvin, “Disseminating Active Map Information to Mobile Hosts,” *IEEE Network*, Vol.8 (5), pp.22–32, 1994.
- [14] E. Shortliffe, B. Buchanan. “A model of inexact reasoning in medicine,” *Mathematical Biosciences*, 23, pages 351–379, 1975.
- [15] M.Uschold, M. Grueninger, “Ontologies: Principles, Methods and Applications,” *Knowledge Sharing an Review*, Vol.11 No.2, 1996
- [16] P. Vajirkar, S. Singh, Y. Lee, “Context-Aware Data Mining Framework for Wireless Medical Application,” *Lecture Notes in Computer Science (LNCS)*, Volume 2736, Springer-Verlag. ISBN 3-540-40806-1, 2003, pp. 381 – 391.
- [17] J. Han, M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
- [18] Office of Government Commerce: Service Support. London: The Stationary Office, 2000.
- [19] Office of Government Commerce: Service Delivery. London: The Stationary Office, 2000.
- [20] OWL (Web Ontology Language) Overview, Available <http://www.w3.org/TR/2004/REC-owl-features-20040210/>
- [21] Protégé, Available <http://protege.stanford.edu/>

[22] Protégé OWL Plugin, Available <http://protege.stanford.edu/plugins/owl/index.html>

[23] Java Server Pages Technology Available <http://java.sun.com/products/jsp/>

[24] DAML, The DARPA Agent Markup Language Available <http://www.daml.org/>

[25] ITIL, OGC Available <http://www.ogc.gov.uk/index.asp?id=2261>

