以本體論為基礎之惡意郵件偵測

姜琇森
雲林科技大學資訊管理學系

施東河
雲林科技大學資訊管理學系

黃信銓
雲林科技大學資訊管理學系

摘要

隨著網際網路的興起，電腦安全成為一個重要的議題，目前普遍使用防毒軟體來防護電腦免於病毒的破壞。這類的防毒機制主要依賴「病毒碼」與「掃毒引擎」的更新才能預防新病毒。根據研究平均每天有 8~10 的新病毒產生，病毒碼更新的防毒方式沒辦法更新病毒碼之前，偵測到新的病毒，系統在新病毒出現而尚未有偵測病毒碼產生的這段期間是非常脆弱且危險的。本論文提出以本體論支援郵件病毒行為偵測及其知識管理的方法，針對郵件病毒的特性建立郵件病毒知識本體，以管理郵件病毒行為相關知識並據以偵測郵件病毒，並根據郵件病毒知識本體間概念與概念之間的關係形態轉換為模糊派翠網路結構進行推論，以偵測郵件病毒。本研究提出智慧型的嵌入式郵件過濾裝置，架設於電子郵件閘道口的郵件安全系統，透過郵件病毒推論引擎，過濾郵件病毒。本研究之電子郵件過濾系統提供友善的 web-based 管理介面，方便管理者進行系統管理及一般使用者來收發信件。

關鍵詞：本體論、模糊派翠網路、郵件病毒、嵌入式系統

Ontology-Based Malicious Email Detection

Hsiu-Sen Chiang
Department of Information Management, National Yunlin University of Science and Technology

Dong-Her Shih
Department of Information Management, National Yunlin University of Science and Technology

Shin-Chuan Huang
Department of Information Management, National Yunlin University of Science and Technology

Abstract

The widespread of Internet causes computer security becomes an important issue. Currently, anti-virus software is the primary mechanism to prevent computers from the damage of virus. Such mechanism relies on the update of virus pattern (or signature) and scan engine to detect a new virus. Eight to ten viruses are created every day and most cannot be accurately detected until signatures have been generated for them. During this time period, systems protected by signature-based algorithms are vulnerable to attacks. We propose a method that uses ontology to support the behavior detection and the knowledge management of email virus. It constructs the ontology of the email virus in accords with the behavior characteristics of the email virus. It then uses the ontology to detect as well as manage the behavior of mail virus. This paper transforms the ontology into fuzzy Petri-Nets to detect the email virus and transforms it into fuzzy Petri-Nets automatically. Finally, we use Protégé 2000 to implement and manage the email virus behavior ontology. We designed and implemented an intelligent email filter with embedded system. It acts as an email gateway to filter inbound messages by enforcing an email virus rule's policies. In the embedded system, we also provided a web-based administrative interface for the system administrators to do the system configuration and to set up their email virus rule filtering policies.

Key words: Ontology, Fuzzy Petri Net, Email viruses, Embedded System

壹、緒論

網路的普及也使得電子郵件的使用情形也大幅增長，慢慢取代傳統的郵件成為新的溝通方式，IDC 估計 2001 年全球平均一天發出的電子郵件數量達 100 億封，預計到 2005 年，數量會加倍成長到 350 億封。電子郵件的盛行衍生出許多問題，其中影響最大的便是電子郵件病毒與垃圾郵件的問題。趨勢科技技術支援部經理林錦忠表示：「Internet 的普及與企業電子郵件的廣泛使用，快速地增加企業競爭力與生產力，但同樣的，也讓電腦病毒找到更快速的傳染媒介。附帶有病毒檔案的電子郵件，往往在短時間內一傳十、十傳百，造成骨牌效應式連鎖感染。國際電腦安全協會 (ICSA) 所公佈的「2005 年度病毒傳播趨勢報告」結果顯示，電子郵件躍升為電腦病毒最主要的傳播媒介，如表 1。換言之，電子郵件已經成為電腦病毒傳遞感染的最主要媒介，如 Happy99、Melissa、求職信、Loveletter、Winbird.a 等都是藉由電子郵件來達到大量傳播的目的，因此如何防範電子郵件夾帶電腦病毒已是不得不被重視的問題。

表 1：病毒傳播媒介比較表　1996-2004 (資料來源：ICSA，2005)

Virus Source	1996	1997	1998	1999	2000	2001	2002	2003	2004
Email Attachment	9%	26%	32%	56%	87%	83%	86%	88%	92%
Internet Downloads	10%	16%	9%	11%	1%	13%	11%	16%	8%
Web Browsing	0%	5%	2%	3%	0%	7%	4%	4%	2%
Other Vector	0%	5%	1%	1%	1%	2%	3%	11%	12%
Software Distribution	0%	3%	3%	0%	1%	2%	0%	0%	0%
Diskette: other	71%	84%	64%	27%	7%	1%	0%	0%	0%

國際電腦安全協會曾估計，平均爆發一隻病毒，至少將波及 25 台電腦，所造成的單位成本損失在 8,000 美元以上。網際網路的發達造成電腦病毒傳播的速度非常快，不在只限於個人或是區域性而是全球性的破壞。郵件病毒所造成的破壞更大，根據趨勢科技的統計，2003 年的 Klez 求職信病毒是第一個歷經一年的變種病毒，造成全球九十億美元受損，超過六百萬台電腦受攻擊；2004 年的 SQL Slammer 病毒，造成全球十億美元受損，超過一百萬台電腦被攻擊。電腦中毒後所造成的損失常常難以估計，系統修復所投入的時間、人力、金錢是個人用戶與企業用戶的痛，無形的效率、商譽、資料的復原，更無法估計。從圖 1 中，更可明顯的看出：當組織遭受病毒感染後，輕則影響個人工作效率；重則造成整個組織營運停擺，所造成的損失不能估計。

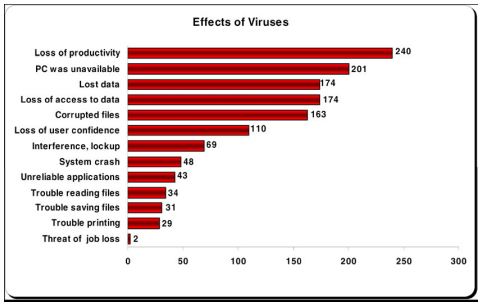


圖 1：病毒在組織內部所造成的損失排名

由於電腦病毒的不斷演變更新，因此近年來電腦病毒的偵測技術也是各方研究的重點，而偵測未知的電腦病毒更是極待被解決的一項重要問題，過往的病毒偵測技術與防毒軟體有一重大缺失，便是等到病毒出現後，再研究病毒特徵，發展出可偵測的病毒碼，如此在新的病毒碼研究出來時，病毒已經造成嚴重的破壞。近來也有相當多的研究是針對未知電腦病毒的偵測，然而在這些方法中一樣也存在著相當多的問題極待克服，如偵測病毒的偵測率、誤判率以及偵測的效能等都是需要解決與探討的問題。

現今的郵件病毒偵測器通常有四個缺點，(1)無論軟硬體價格皆非常昂貴，動輒數十萬元，對企業、組織或個體而言成本都相當高，(2)在病毒偵測過程中，防毒軟體佔用太多系統資源，造成使用者使用上極大的不便，(3)防毒軟體必須靠使用者自行更新病毒碼，才能確保自己的系統處於安全的狀態，對大部分的使用而言，並沒有這樣的觀念，(4)未能偵測未知、新的病毒。加上現今病毒的傳播非常快速，並迅速卸載沒有更新的防毒軟體，釀成巨大的損失。

本研究期望透過分析郵件病毒行為中無法更改的特徵，根據這些特徵行為來偵測新的、未知的病毒，如此則不必定期更新病毒碼，也能有效偵測出病毒，並將病毒所造成的傷害降到最低。由於知識本體論能將知識做適當的呈現與表示其應用面非常廣泛，本研究擬分析病毒的行為模式，透過本體論 Ontology 知識表示的特性有效的呈現病毒行為特徵，建立郵件病毒行為的本體論作為過濾規則的依據，並運用 Petri Net 的流程判斷模式，將郵件病毒行為本體轉換為病毒偵測的推論引擎，有效過濾已知與未知的郵件病毒，將損害降至最低。本研究提出之郵件病毒過濾架構採用嵌入式系統配合 Linux 平台，能有效降低系統成本，且本架構之設計，能讓使用者透過適當的介面登入嵌入式系統瀏覽信件，而不用將信件下載到個人主機，這樣的設計除能提高安全性外，更能避免佔用系統資源的問題，讓使用者充分運用系統效能。

貳、文獻探討

本研究針對郵件病毒、病毒偵測方法、嵌入式系統的相關文獻加以探討，藉以歸納出電腦病毒與郵件病毒所具有之特性，並檢視目前偵測方法所存在的問題。

一、郵件病毒

郵件病毒在執行破壞行動時，會取得該系統寄送郵件的權力，將病毒本身複製後並郵寄遞送出去(Kienzle and Elder, 2003)，劉順德 (2001)定義凡透過電子郵件作為傳播管道的病毒(Viruses)或蠕蟲(Worm)，均稱為郵件病毒，Takeshi and Yoshiteru (2002)則認為郵件病毒就是在病毒感染系統後，不靠使用者操作系統(如檔案複製，檔案傳輸等)來散播病毒，而以電子郵件作為其傳播媒介。因此郵件病毒就是不管任何種類的電腦病毒，如特洛伊木馬、蠕蟲、病毒等，感染或侵入系統後，以電子郵件作為其傳播的媒介，將病毒本身向網路其他主機散播出去。

Kienzle and Elder (2003)的研究指出，早期的郵件病毒比較單純，利用系統本身的郵件軟體或呼叫執行郵件 API 的程式，將自我複製的病毒碼寄送到單一或多個郵件位址。其後的郵件病毒則逐漸演變成擁有自己的 SMTP 程式，不再只依靠系統的郵件程式來達到其散播的目的，近年來，郵件病毒更已經開始利用人們轉寄電子郵件的習慣，附加一些欺騙的手段來引誘使用者上當，進一步的協助病毒散佈。

（一）郵件病毒感染的方式

電子郵件主要可以分為兩個部分，一是郵件本文，二是附件夾帶檔的部分，因此郵件病毒要透過電子郵件來散播傳染，其感染的方式主要有兩種：

１、透過附加檔案

寄發電子郵件時，使用者可以夾帶任何形式的檔案一起寄送，因此電腦病毒只要能隱匿在這些夾帶檔案之中便可以藉著電子郵件而散播到網路之中，這種方式也是一般較為傳統的方式，由前述電腦病毒的特性中大部分的電腦病毒都可以隱匿在檔案之中。

２、透過郵件本文

郵件本文的格式主要可以分為純文字格式與 HTML 格式，在純文字格式下的郵件本文比較不具潛在的威脅。由於網際網路的盛行，新種的電腦病毒具備藏匿於網頁之中的能力，因此 HTML 格式的郵件本文中較容易會有電腦病毒隱匿其中，再加上有些郵件收信軟體會自動瀏覽郵件本文，所以容易在不自覺之中觸發了電腦病毒而造成感染。郵件病毒採用不同的技術造成上述兩種傳播模式，利用 SCRIPT 語法、MIME 格式上的漏洞、Java applet 等技術內嵌於郵件本文中，利用附加惡意的檔案格式來傳播。

（二）郵件病毒的偽裝方式

姜琇森 (2003)觀察指出郵件病毒為了隱匿自己本身、引誘使用者上當或者要欺瞞使用者使其降低警戒的心，使用各種偽裝的方法，以下將針對這些方法概略的介紹。

１、郵件標題與內容的改變

利用一些人類心理上的觀念或弱點，將郵件的標題與內容改變或互相搭

配，以降低收件者的警戒心或引發收件者的好奇心，尤其是特別的日子(聖誕節或情人節)或特別的事件(911、總統大選)，讓收件者開啟信件或執行附加檔案，使收件者中毒。

2、盜用寄件者名稱

郵件病毒感染電腦後，會附至其郵件通訊錄中的名單或取得郵件紀錄採回信的方式，都讓收件者以為是朋友或熟人寄來的信，在毫無防備的情況下，導致系統中毒。

3、改變夾帶檔副檔名

windows 系統根據副檔名啟動開啟執行的應用程式，某些相關的副檔名可被直接執行或是被同一應用軟體所執行，因此將具有執行程式能力的副檔名改變成不常見的檔案格式，以降低使用者的警覺心，進而使其中毒。但更改副檔名需配合相同類型的檔案類型才能生效，常見的組合如下：XLS 及 DOC：這二種檔案格式都是屬於MSOffice提供的檔案格式，主要是利用Office巨集的共通性，使得病毒可以在這兩種檔案格式上做轉換。

4、偽裝副檔名

為了欺騙使用者，郵件病毒會偽裝附加檔案的附檔名，來降低使用者的警覺心，大部分的病毒會使用雙重副檔名的方式，如 readme.txt.exe 或 goldfish.doc.pif，讓粗心的使用者以為這是個純文字檔，其實是 EXE 可執行檔，著名的 Nimda 病毒也是採用這種方式。

5、隱匿在附加檔案中

這種欺騙方式通常會和郵件標題與內容相配合，收件者在執行附加檔案後，由於有其他程式的執行效果作掩飾，故收件者不會對執行的結果有任何異常的感覺。更進而轉寄給自己的朋友，如聖誕節病毒(WORM_MALDAL.C)即以 Flash 動畫作掩飾。

（三）郵件病毒傳播能力

劉順德 (2001)的研究指出郵件病毒的傳播能力可分為兩種，一種為大量傳播型，另一種為非大量傳播型，分別闡述如下：

1、大量傳播型的郵件病毒

依通訊錄的名單：透過被感染的目標機器中的通訊錄名單，來決定下一個傳播目標。這種方式是最常見的。

2、非大量傳播型的郵件病毒

這類的郵件病毒較前者精巧，其目的並不在於傳播點廣，而在於傳播的隱密性。現有的郵件病毒以這種方式傳播的並不多，例如 W32.Navidad 即為一例。這類的郵件病毒會監視受感染電腦的郵件行為，利用正常的郵件行為

來夾帶郵件病毒至下一個傳播點。常見的手法，是透過替換 windows 系統中的 winsock.dll 檔，來達到監視與修改郵件的目的。著名的 Happy99.Worm 即為一例。

二、病毒偵測的方法介紹

1987 年，電腦病毒的問題首次被重視，自此之後電腦界發展了數種電腦病毒的偵測方式，根據趨勢科技、賽門鐵客、瑞星科技防毒中心表示，目前的病毒偵測方法大約有下列幾種：病毒碼比對法(matching virus definition patterns)、加總比對法(check sum)、即時 I/O 掃描法(real time I/O scan)、行為分析法(Behavior-Based virus detection)、智慧代理者偵測法(Agent-Based virus detection)等，以下將介紹這一些病毒偵測方式(Zenkin, 2001；Lee et al., 1997)。

（一）病毒碼比對法

病毒碼是指病毒中一段具有特色或是較為特殊的程式碼，也就是所謂的病毒特徵，病毒碼就像是我們每個人都有特定的指紋與 DNA 一樣，是辨別電腦病毒的重要依據，病毒碼掃描法就是建立起每隻電腦病毒的病毒碼資料庫，每當執行偵測掃描程式時，就會根據病毒資料庫中的病毒碼與程式檔案作比對，檢測是否有電腦病毒存在於檔案之中。病毒碼掃描法最大的優點是執行效能佳，偵測結果準確，只要是已存在病毒資料庫的病毒一定會被偵測出來，目前市面上大多數的防毒軟體均採用這種偵測掃描方式，但其缺點是只能偵測出已知的電腦病毒(Phillippo, 1990; Cohen, 1991)，當病毒碼資料庫沒有該病毒資料時則無法偵測到未知的新病毒及變種的病毒(Zenkin, 2001)。

（二）加總比對法

這個方法主要是利用特定的演算法檢驗檔案的完整性，因為大多數的病毒會依附或寄生於其他的程式，所以被感染的程式會有檔案大小增加的狀況產生或是檔案日期被修改的情形。因此根據每個程式檔案的名稱、大小、時間、日期及內容，加總為一個檢查碼，再將檢查碼附於程式的後面，或是將所有檢查碼放在同一個資料庫中，再利用此 Check-sum 系統，追蹤並記錄每個程式的檢查碼是否遭更改，以判斷是否中毒。這種技術可偵測到各式的病毒，但最大的缺點就是誤判高，且無法確認是哪種病毒感染的，而這樣的偵測方法也限制了作業系統本身的彈性，反而變得不切實際(Luke and Harris, 1999)。

（三）即時輸出/輸入掃描法

即時的 I/O 掃描的方法主要是即時地攔截資料的輸入/輸出動作，並且對這些資料流做病毒碼掃描，希望能夠在病毒尚未被執行之前，就能夠防堵下來。這種方法取代了傳統病毒碼掃描特定時間掃毒的缺點，但整體的病毒掃描機制與病毒碼掃描法是相同的，因此這個方法一樣對未知的電腦病毒不具有效的防護能力，而且這樣攔截系統所有輸出/入動作並作即時掃描，會對系統的資料傳輸效率有一定程度的影響存在(Phillippo, 1990)。

（四）行為分析法

行為分析法是利用病毒的特有的行為特徵或偵測電腦中一些不正常的行為及指令來監測病毒的方法。通過對病毒多年的觀察、研究，有一些行為是病毒的共同行為，而且比較特殊。在正常程式中，這些行為比較罕見。當程式運行時，監視其行為，如果發現了病毒行為，立即報警。一些防毒軟體利用分析檔案中一些不正常的動作來作病毒偵測，例如無預警的格式化磁碟、同時執行許多執行緒、或是刪除系統資料夾中的檔案等等。行為分析法可發現未知病毒、可相當準確地預報未知的多數病毒。但它不能識別病毒名稱，而且在軟體實現時有一定的難度。

賽門鐵克的 Bloodhound 的技術改進了上述的方法，會先分析出病毒可能在程式中藏身的地方，接著分析這段程式區間(program region)的程式邏輯(program logic)判斷該程式可能產生的動作，經由這些動作判斷這段的程式區間是否會對系統造成損害，並依判斷結果來決定是否要發出警訊。趨勢科技的 ScriptTrap 掃瞄技術會捕捉病毒感染檔案時所發出的指令，並可識破 JavaScript 與 VBScript 病毒變種或嵌入程式碼至 HTML、XML 等任何偽裝手法，利用病毒模式檔案中已知的病毒活動資料庫來比對這些獨立元素，若發現吻合，該程式碼將被宣告為變種，掃瞄引擎隨即檢測出該病毒。此外，許多研究使用決策樹與 Naive Bayes、SOM 與 K-Medoids 等資料探勘方式來偵測未知病毒(Shih et al, 2004；2005)。

（五）以代理人程式為基礎的偵測法

這個方法主要的目的是在偵測未知的病毒及智慧性病毒(intelligent virus)。由於病毒破壞的目標通常以作業系統為主，所以代理人程式(agent)本身很少直接受到病毒的威脅，且代理者機動(Mobile)的特性可以在網路上需要的地方設置，所以近年來許多知名的廠商如 IBM 也朝這個領域深入研究。日本 T. Okamoto 等學者則是提出另一種作法，其設計多重異質主機代理人(heterogeneous agents)的方法，這些異質主機是經由網路連結起來，透過這些異質性代理人互相檢查所在主機的檔案是否遭到感染（亦即遭到修改），若是的話，則會從其他沒被感染的主機將檔案復原，換句話說，其將病毒偵測與防毒的機制變成一個檔案備份與還原的動作。由於代理人程式仍是藉由分析檔案來判斷是否存在病毒，所以這種方法遇到一些特殊的病毒就無法發揮其效用了，例如傳播速度快的病毒。

隨著電腦技術不斷的進步，電腦病毒與反病毒的技術之間的較量永遠也不會停止，現今的電腦病毒已不再像從前一樣的單純，通常都是複合多種病毒或蠕蟲的特性，以各種偽裝隱匿的方式來閃避防毒軟體的偵測，因此防毒技術不應再侷限於單一的方法，傳統的病毒碼偵測、掃描法，甚至啟發式分析法等防毒技術都過於注重靜態特徵的偵測，而現今郵件病毒具備複合、傳播快速與自行變種的特性，且蘊含複雜的行為，並搭配不同的偽裝與欺騙手法，使得過去方法的誤判率也隨之提升。對於現今病毒的特性與發展，過去的偵測方法成效有限，且佔用的相當大的系統資源。

目前針對未知病毒的偵測技術已朝病毒不可變更的特徵著手，為對抗病毒複合與多變的特性，病毒行為特徵偵測已是目前主流。故本研究利用本體論(Ontology)能有效呈現知識本體的特性，將郵件病毒行為流程與運作適當的表示，運用模糊派翠網路(Fuzzy Petri Net)能描繪狀態轉換的功用，模擬出郵件病毒多變的行為特徵，形成模糊規則(Fuzzy Rule)，針對各種不同情況與行為，作出最適當的推論。面對現今郵件病毒多變、複合與傳播快速的複雜特性，本體論能具體呈現其複雜的關係，而模糊派翠網路(FPN)依據不同的情況，透過模糊推論過程作適當的判斷，有效的偵測出病毒，並降低誤判的可能性。

三、嵌入式系統

在嵌入式系統應用範圍非常廣泛，如應用在微處理機控制器、信號處理器、工業用電腦、資訊家電、和 PDA 等。常用的嵌入式硬體如 x86 架構、ARM 架構、MIPS 架構等。目前常見並相容於 Linux 技術的處理器主要分為三大類：

- ARM：ARM7,ARM9,Xscale,StrongARM

ARM 系列 CPU 中最常見的是 Intel 公司的 StrongArm CPU，現階段被廣泛的應用在 Handheld PC，例如：COMPAQ 的 iPAQ、CASIO 的 CASIO Cassiopeia E-115、國眾的 LEO FreeStyle E300 等等。

- PowerPC:4xx,7xx,9xx

PowerPC 系列 CPU 中當推 Palm 所採用的 Motorola 公司之 DragonBall 系列 CPU，Palm 的相容機種應該是大家最耳熟能詳的，例如：Palm V 系列、m 系列，Handspring 的 Visor 系列，令人驚豔的 Sony CLIE PEG 系列 PDA。

- x86:x86 架構的嵌入式系統目前也很普及

這邊指的 x86 系列 CPU 是針對嵌入式系統為主，如美國國家半導體(NS)公司的 Geod 系列 CPU。

ARM 微處理器與市場上其他微處理器的最大的差異在於 ARM 創造了低成本及低耗電的 RISC 架構，而其他的架構則著重於提高性能的微處理器.以 ARM 為核心的應用晶片，產品種類遍及 PDA，數位相機，藍芽，VoIP(Voice over IP)網路處理器(networking processor)，機上盒(Set-Top Box)，行動電話(尤其是 Smart Phone)等各式各樣的電腦週邊，消費性電子及通訊產品。此外，ARM 也獲得許多即時作業系統(Real Time Operating System, RTOS)的支援，比較知名的有 Windows CE, Linux, pSOS, VxWorks, EPOC, uCOS, BeOS 等，在嵌入式產品的應用領域更是無往不利。

x86 跨進嵌入領域應用，並非與原有的嵌入式市場爭搶地盤，例如許多雷射印表機使用 AMD 29000 處理器，或者許多網路交換器使用 MIPS 核心處理器，或磁碟陣列卡使用 Intel i960 的 I/O 處理器，在這些既有領域中 x86 處理器皆無切入優勢，x86 處理器的嵌入式應用多半是在新轉化型態的硬體上有所發揮。在嵌入式 CPU 領域中，掌握嵌入式 CPU 核心技術的公司有 Transmeta、威盛(VIA)、ARM 和 MIPS 等。其中 ARM、

MIPS 的 CPU 採用全新的架構，與這類的公司合作，意味著軟體系統也將受其專利技術的限制。而威盛、Transmeta 的 CPU 採用的是 x86 或相容的架構，可以很方便地利用現成流行的作業系統和應用軟體，在技術少了一道發展的屏障。

本研究之嵌入式系統平台為晶慧公司生產的 NET-Start-IXP!。該發展板中央處理器核心為 32-bit Intel XScale–IXP420，IPX420 在應用方面是使用高效能的 RISC 處理器，它被廣泛應用在網路上，並且提供了許多的擴充介面及標準的 PCI 介面，例如：8MB Flash、32MB SDRAM、LAN port*4、WAN port、16MB NAND-Flash、USB 等。軟體方面，在眾多的作業系統中，我們使用的是開放原始碼的 Linux 平台，作業系統為 BusyBox，並且安裝上同樣是開放原始碼的軟體–Apache+PHP Server 做為郵件病毒過濾器之伺服器。

參、研究方法

本研究擬採用本體論(Ontology)，表示與呈現郵件病毒行為的知識本體，描繪與敘述各行為本體內部類別(Class)與樣式(Individual)的概念，以及之間相配合的屬性(Property)關係，運用模糊派翠網路(Fuzzy Petri Net)圖形塑模的能力，將郵件病毒知識本體轉換成行為網路，期望建立最佳判斷規則庫，再利用模糊理論的特性，定義各行為特徵的隸屬函數(Membership Function)，透過模糊推論方法依循行為特徵網路判斷電子郵件是否歸屬於郵件病毒。

一、本體論(Ontology)

Ontology 在哲學的領域中已經存在很久，在學者 Bunge (1977)對此定義是：「關於真實世界的基本特性」，但是現在有許多學者利用此特性，將此應用在電腦科學領域上，例如知識工程（knowledge engineering）、知識表達（knowledge representation）、定性塑模（qualitative modeling）、語言工程（language engineering）、資料庫設計（database design）、資訊塑模（information modeling）、資訊整合（information integration）、物件導向分析（object-orient analysis）、資訊檢索與存取（information retrieval and extraction）以及知識管理與組織（knowledge management and organization）等(Guarino, 1998)。

（一）Ontology 之定義

Ontology 這個字最早是從哲學而來，其意義是解釋現實存在的基本特性，而這個字漸漸被用於知識工程方面，最早是由 Neches et al. (1991)所定義，認為 ontology 是定義基本詞彙(terms)以及關係(relations)，並且由主要領域的字彙(vocabulary)所組成，此外，利用結合詞彙與關係的規則，更可定義廣泛的字彙。Swarout et al. (1997)說明 Ontology 是一種有階層架構之詞彙集合，而這些詞彙是可以描述領域知識，並且被用在知識庫中的基礎架構。Guarino (1998)定義 Ontology 是邏輯原理的集合，並且設計

成可用來說明字彙的深層涵義。

總結上述的定義，ontology 是利用邏輯理論，將領域中的知識轉換成概念化的規則，其中知識是包含詞彙與之間關係，並且利用階層架構以及正式化的語言呈現，以提供做知識的共享或是再利用。

（二）Ontology 之特性

Chandrasekaran et al. (1999)表示所有理論可以分為兩大類，一種是機械理論(Mechanism Theories)，另一種是內容理論(Content Theories)。而 Ontology 是一種內容理論，是可以在特殊領域知識之中，呈現物件、物件屬性、物件之間可能的關係，並且有下列特性：

- Ontology 是一種字彙(Ontology as vocabulary)

Ontology 在哲學中主要探討存在的事物，而在人工智慧上是探討兩個事務之間的相關意義。第一，Ontology 可視為一種表達的字彙，最常為人用來做特殊領域或論題的探索。第二，通常利用 Ontology 來描述某領域的知識，換言之，Ontology 提供了一套描述領域真象的述語。

- Ontology 是內容理論(Ontology as content theories)

在人工智慧研究社群中，一個好的機制，類神經網路、模糊邏輯等機制，不能沒有一個好的領域內容理論來支援，以及需要呈現的語言。而 Ontology 是一個內容理論，因為它的主要貢獻在找出物件的規格型態之間的關係。黃崇益 (2002)整理出 ontology 的三個特性，第一，可描述人世間所有事物；第二、達到語法的互通性(Syntactic Interoperability），第三、達到語意的互通性（Semantic Interoperability）。

由以上特性可以了解，ontology 不但可以呈現領域知識的本質，而且可以詳細描述知識內容的概念、屬性，以及概念之間的關係。因此，可以透過 ontology 的理論基礎，來呈現知識結構，以可完整且清楚描述領域知識內容。

（三）Ontology 之建立方法與方法論

Uschold and King (1995)是第一個提出明確的 ontology 建置方法與方法論，之後陸續有 TOVE (Gruninger and Fox, 1995)、KACTUS (Bernaras et al., 1996)、METHONOLOGY (Gomez-Perez et al., 1996)、SENSUS(Swarout et al., 1997)以及 On-To-Knowledge(Staab et al., 2001)。而不同方法論都有其特色並應用在不同領域知識上，因此利用 Fernandez-Lopez et al. (1997)所列 8 項衡量指標分析 ontology 的方法論，分析其方法論之特性是否符合標準。不同領域可以依照領域獨特的知識特性選擇最適合的方法論，例如，利用 METHONOLOGY 建立化學 ontology，其中結合化學元素(包含 16 個 classes 以及 103 個 instances)與化學晶體(包含 19 個 classes 以及 66 個 instances)兩種 ontology，呈現化學領域的知識架構以及關係(Fernandez-Lpoez et al., 1999)。

本研究擬採用本體論(Ontology)強大的知識表示與呈現能力，描繪與敘述郵件病毒本體的各種行為，以及之間相配合的行為屬性關係，以建構郵件病毒行為的知識本體，成為建構模糊派翠網路(Fuzzy Petri Net)的基礎知識依據。

二、模糊派翠網路（Fuzzy Petri net）

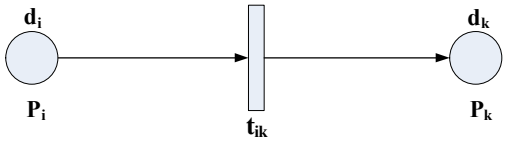
Petri net 理論，簡稱 PN，於 1962 年由德國數學家 Carl Adam Petris 提出，是一個圖形化與數學模式的模組工具，可以描述與研究資訊流程系統(Murata, 1989)。而 PN 有別於一般的圖形塑模工具，是屬於動態模組(Dynamic modeling)，除了具備適用於表現平行與分散式系統之動態行為，簡單且階層式的圖示方法，亦提供數學理論以供系統作定性或定量的分析，更可進一步地發展在特殊流程上面。模糊派翠網路，簡稱 FPN，是將模糊推論過程融入派翠網路之中的一種高階派翠網路結構，可架構在原有規則知識庫系統中，用以產生新的模糊知識規則並可描述其推論過程，藉由模糊派翠網路所產生之模糊產生規則(Fuzzy production rule, FPR)，將透過其特有之轉移節點推論機制，推論出輸入/輸出狀態節點間之因果關係。表 2 為模糊派翠網路模組符號定義，圖 2 為包含一轉移節點及兩個狀態節點之基本模糊派翠網路模組，並圖例說明各組成元素之表示法：

表 2：Fuzzy Petri net 之定義

一個 FPN 有八個變數值，FPN = (P、T、D、I、O、f、α、β)：
P = { $p_1, p_2, \ldots, p_m$ }為狀態節點所構成的有限集合，m 表示在模糊派翠網路中狀態節點的個數，$m \geqq 1$。
T = { $t_1, t_2, \ldots, t_n$ }為轉移節點所構成的有限集合，n 表示在模糊派翠網路模組中轉移節點的個數，$n \geqq 1$。
D = { $d_1, d_2, \ldots, d_m$ }為狀態節點中之狀態描述(propositions)所構成的有限集合，m 表示在模糊派翠網路中狀態描述的個數，$m \geqq 1$。
I：P→T 為輸入函數，表狀態節點到轉移節點間之線段。
O：T→P 為輸出函數，表轉移節點到狀態節點間之線段。
f：T →[0, 1]為轉移節點中數值之關聯函數(association function)，值介於 0～1 之實數。
α：P →[0, 1]為狀態節點中數值之關聯函數(association function)，值介於 0～1 之實數。
β：P→D 為狀態節點中文字描述之關聯函數(association function)。

FPN 有三個基本組成單位：(1)狀態節點(place)，為圓形。(2)轉移節點(transition)，為長條形。(3)方向弧(arcs)，為箭號，是連接狀態節點與轉移節點。FPN 理論最主要是在於動態流程之探討，因此其執行規則是在強調轉變的能力(enable)與轉變觸發點(fire)，也就是觸發(fire)有能力(enable)的轉移節點(transition)，而在狀態轉變時，根據不同條件狀態觸發而加入模糊推論的能力，進一步探究影響前狀態對後狀態的影響與關係程度。FPN 是種特殊的直接圖形，內含一個初始情況，可以稱為初始狀態(initial marking)，M0。假設 N 為一個 FPN，則 N 會有狀態節點(place)與轉移節點(transition)兩種節點，而方向弧(arcs)是從一個狀態節點(place)關係到一個轉移節點(transition)或者是從一個轉移節點(transition)關係到一個狀態節點(place)，並且方向弧(arcs)會給予一個權重(正整數)，而轉移節點(transition)包含一個模糊關聯函數 f，將不同前狀態影響後狀態的關係轉換成適當的表示值。狀態(marking)是指派一個非負整數，亦表標誌

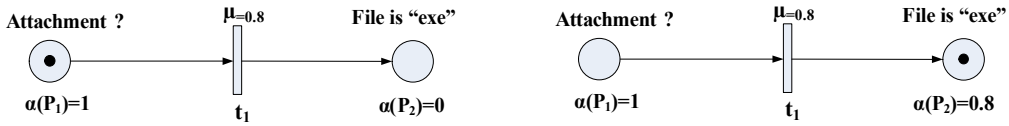
(token)，到每一個狀態節點(place)，假如狀態(marking)是指派一個非負整數 y 到狀態節點(place) p，表示 p 有 y 個標誌(token) (Chen et al, 1990；Murata, 1989)。



FPN = (P, T, D, I, O, f, α, β)
$P=\{P_i, P_k\}$, $D=\{d_i, d_k\}$, $T=\{t_{ik}\}$
$I(t_{ik})=\{P_i\}$, $O(t_{ik})=\{P_k\}$
$\alpha(P_i), \alpha(P_k) = [0\sim1]$
$\beta(P_i)=d_i$, $\beta(P_k)=d_k$

圖 2：模糊派翠網路定義表示圖

模糊派翠網路之推論機制是藉由不同型態之模糊產生規則(FPR)，根據不同狀態節點輸入，考慮狀態節點與轉移節點間模糊關聯函數，透過特有的模糊推論方式，推論下一狀態節點的可能狀態。圖 3a 及圖 3b 分別表單一輸入狀態節點觸發前後之模糊生產規則，當轉移節點 t_1 觸發後，其輸出節點之模糊數值計算，如公式(1)所示。



(a)觸發前　　(b)觸發後

圖 3：Type 1 觸發前後之模糊生產規則

其中，轉移節點 t_1 被觸發前，P = { p_1, p_2 }；T = { t_1 }；D = { Attachment ?, File is exe}；$I(t_1)$ = { p_1 }；$O(t_1)$ = { p_2 }；$f(t_1) = \mu_1 = 0.8$；$\alpha(p_1) = 0.9$；$\alpha(p2) = 0$；$\beta(p_1)$ = { d_1 } ="Attachment ?"；$\beta(p_2)$ = { d_2 } ="File is exe"，轉移節點 t_1 被觸發後，$\alpha(p2) = 0.8$，代表 $\beta(p_1)$ = { d_1 } ="Attachment ?"前狀態節點對 $\beta(p_2)$ = { d_2 } ="File is exe"後狀態節點造成影響的可能程度為值 0.8。

（一）模糊產生規則(Fuzzy production rule)

模糊派翠網路中考慮推論過程的不確定因素，將模糊邏輯處理不確定性方法加入派翠網路的推論機制。將模糊值加入派翠網路之後，產生五種不同形式模糊生產規則(Fuzzy production rule；FPR)，其推論過程，如下所述(Chen et al., 1990)：

Type 1: IF d_j THEN d_k (CF= μ). 這種形式的模糊生產規則其推理過程可以透過公式(1)的形式來描述

$$\alpha(P_k)=\alpha(P_j)*\mu \tag{1}$$

Type 2: IF d_{j1} and d_{j2} and….and d_{jn} THEN d_k (CF= μ). 這種形式的模糊生產規則其推理過程可以透過公式(2)的形式來描述

$$\alpha(P_k) = \min\{\alpha(P_{j1}), \alpha(P_{j2}), \ldots \alpha(P_{jn})\} * \mu \quad (2)$$

Type 3: IF d_j THEN d_{k1} and d_{k2} and….and d_{kn} (CF=μ). 這種形式的模糊生產規則其推理過程可以透過公式(3)的形式來描述

$$\alpha(P_{k1}) = \alpha(P_j) * \mu, \alpha(P_{k2}) = \alpha(P_j) * \mu, \ldots \quad \alpha(P_{kn}) = \alpha(P_j) * \mu \quad (3)$$

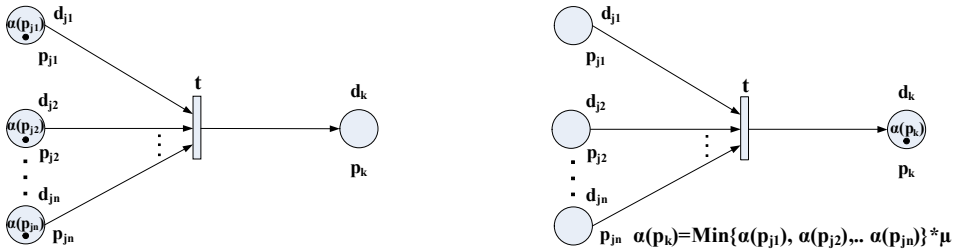
Type 4: IF d_{j1} or d_{j2} or….or d_{jn} THEN d_k (CF= μ_i). 這種形式的模糊生產規則其推理過程可以透過公式(4)的形式來描述

$$\alpha(P_k) = \max\{\alpha(P_{j1}) * \mu_i, \alpha(P_{j2}) * \mu_i, \ldots \alpha(P_{jn}) * \mu_i\}, i = 1,2,\ldots n \quad (4)$$

Type 5: IF d_j THEN d_{k1} or d_{k2} or….or d_{kn} (CF= μ_i). 這種形式的模糊生產規則其推理過程可以透過公式(5)的形式來描述

$$\alpha(P_{k1}) = \alpha(P_j) * \mu_i, \alpha(P_{k2}) = \alpha(P_j) * \mu_i, \ldots \alpha(P_{kn}) = \alpha(P_j) * \mu_i, i = 1,2,\ldots n \quad (5)$$

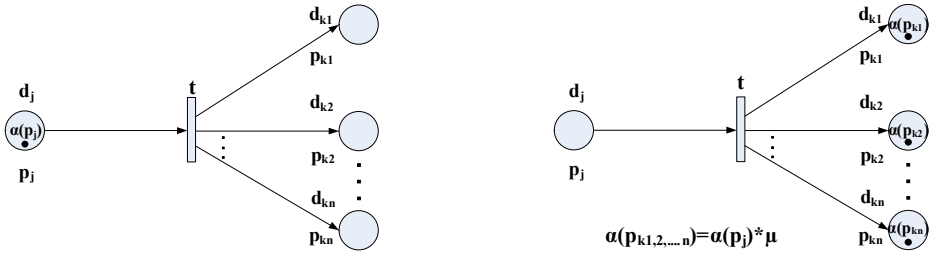
圖 4a 及圖 4b 分別表觸發前後之 Type 2 的模糊生產規則，當轉移節點 t 觸發後，其輸出節點之模糊值計算，如公式(2)所示。



(a)觸發前 (B)觸發後

圖 4：Type 2 觸發前後之模糊生產規則

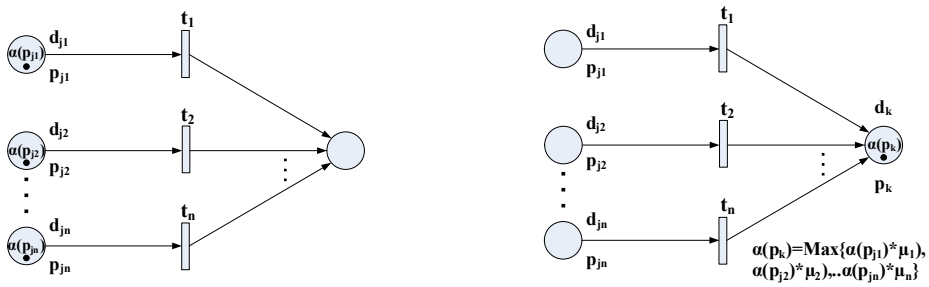
圖 5a 及圖 5b 分別表觸發前後之 Type 3 的模糊生產規則，當轉移節點 t 觸發後，其輸出節點之模糊值計算，如公式(3)所示。



(a)觸發前 (b)觸發後

圖 5：Type 3 觸發前後之模糊生產規則

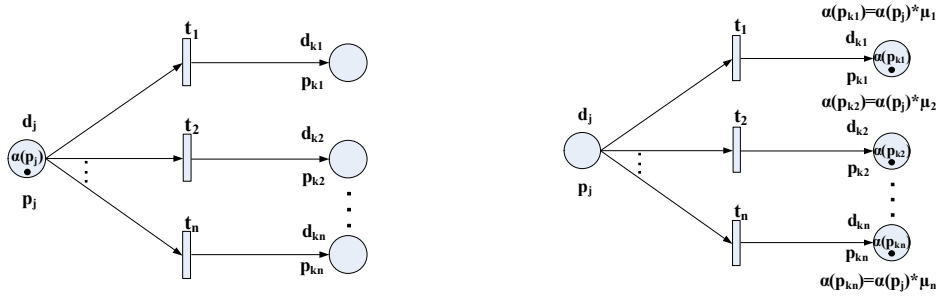
圖 6a 及圖 6b 分別表觸發前後之 Type 4 的模糊生產規則，當轉移節點 $t_1, t_2, \ldots, t_n$ 觸發後，其輸出節點之模糊值計算，如公式(4)所示。



(a)觸發前 (b)觸發後

圖 6：Type 4 觸發前後之模糊生產規則

圖 7a 及圖 7b 分別表觸發前後之 Type 5 的模糊生產規則，當轉移節點 $t_1, t_2,\ldots,t_n$ 觸發後，其輸出節點之模糊值計算，如公式(5)所示。



(a)觸發前 (b)觸發後

圖 7：Type 5 觸發前後之模糊生產規則

（二）模糊派翠網路的推論演算法

模糊派翠網路之推論係利用派翠網路之並行處理特性，從起始狀態節點至目的狀態節點的多種不同路徑中，選擇一條最佳路徑，以提升推論之速率及準確性，因此，模糊派翠網路模型之初始條件，除初始標記外，亦包含起始、目的(start \ gaol)兩狀態節點選定。直接可到達集合(immediately reachable set, IRS)及可到達集合(reachable set, RS)如圖 8 為兩個轉移節點 t_1, t_2 及三個狀態節點 p_1, p_2, p_3 所構成之基本模糊派翠網路模型，則狀態節點 p_1 透過轉移節點 t_1 之激發，可將其狀態直接轉移至狀態節點 p_2，稱為 p_1 直接到達 p_2，亦可寫成 IRS(p_1) = { p_2 }。若狀態節點須透過兩個或多個轉移節點之激發，方可將其狀態直接轉移至目標狀態節點，則稱為可到達，如狀態節點 p_1 透過轉移節點 t_1, t_2 之激發，可將其狀態直接轉移至狀態節點 p_3，稱為 p_1 直接到達 p_2 亦可寫成 IRS(p_1) = { p_2 }；p_2 直接到達 p_3 亦可寫成 IRS(p_2) = { p_3 }。而 p_1 之可到達集合(RS)則為 RS(p_1) = { p_2, p_3 }(Chen et al, 1990)。

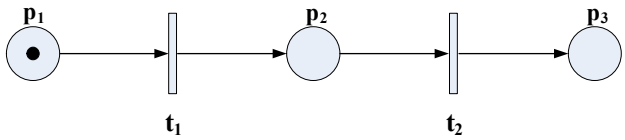


圖 8：模糊派翠網路 IRS 及 RS 表示圖

模糊派翠網路之各狀態節點是否擁有標記，係依狀態節點中之模糊數值大小而定，若模糊數值 $\alpha(p_i)$大於或等於所設定之起點數值(threshold value) λ，即 $\alpha(p_i) \geqq \lambda$，則稱狀態節點 p_i 取得標記，反之則 p_i 無法取得標記。模糊派翠網路之推論係利用派翠網路之並行處理特性，從起始狀態節點至目的狀態節點的多種不同路徑中，選擇一條最佳路徑，其推論演算法分述如下：

INPUT: (a)起始節點與各節點之模糊程度值 $\alpha(p)$，$\alpha(p) \in [0, 1]$。

(b)起始節點與各節點對應之狀態 $\beta(p)= d$。

(c)起始節點與各節點之門檻值 λ，$\lambda \in [0, 1]$。

(d)各轉移節點之確定因素關聯函數 CF。

(e)各轉移節點之確定因素值 CF = μ，$\mu \in [0, 1]$。

OUTPUT:終端節點表達狀態之模糊程度值，$\beta(p_g)= d_j$。

a. 初始條件輸入

起始狀態節點 p_s 必須擁有標記，即 $\alpha(p_s) \geqq \lambda$，使文字描述狀況 $\beta(p_s)$為真。

b. 狀態資料傳遞

依照下列四步驟即可完成由起始狀態節點 p_s 至目的狀態節點 p_g 之生長樹(Sprouting tree)建立，作為最佳路徑選擇推論之依據。

步驟一：

(a)必須擁有標記。

(b)必須不是終端節點， 並紀錄成$(p_s, \alpha(p_s), \mathrm{IRS}(p_s))$，作為生長樹之起點。

(c)透過轉移節點 t_s 將起始狀態節點 p_s 之狀態傳遞至 $\mathrm{IRS}(p_s)$，其 $\mathrm{IRS}(p_s)$之模糊數值可經由 $\alpha(\mathrm{IRS}(p_s)) = \alpha(p_s) * f(t_s)$計算得知，並紀錄為$(\mathrm{IRS}(p_s), \alpha(\mathrm{IRS}(p_s)), \mathrm{IRS}(\mathrm{IRS}(p_s)))$。

步驟二：生長樹之建立敘述如下：

(a)終端節點：若某一狀態節點之直接到達節點為空集合，狀態節點 p_e 及 p_y，可寫成 $\mathrm{IRS}(p_e) = \{\phi\}$或 $\mathrm{IRS}(p_y) = \{\phi\}$，則稱該狀態節點為終端節點。終端節點將不被列入生長樹中。

(b)非終端節點且無相鄰之狀態節點：

目標狀態節點 $p_j \in \mathrm{IRS}(p_i)$，且 $\alpha(p_i) \geqq \lambda$，且轉移節點 t_{ij} 之確定因素 $\mathrm{CF}_{ij} = \mu$，其中，$\mu \in [0, 1]$，則可由既有之生長樹節點$(p_i, \alpha(p_i), \mathrm{IRS}(p_i))$以一條線段連接到新建立節點$(p_j, \alpha(p_j), \mathrm{IRS}(p_j))$，並於線段旁標示 μ，且於新建立之節點旁標示「達成節點」，即表示該條由 p_i 到 p_j 之路徑完成。

(c)非終端節點且擁有一個以上之相鄰狀態節點：

若狀態節點 p_u 與 p_a、p_b、…、p_n 於轉移節點 t_{uv} 上為相鄰，且目標狀態節點 $p_v \in \mathrm{IRS}(p_u)$，且 $\alpha(p_u) \geqq \lambda$，且轉移節點 t_{uv} 之確定因素 $CF_{uv} = \mu$，其中，$\mu \in [0, 1]$，則可由既有之生長樹節點$(p_u, \alpha(p_u), \mathrm{IRS}(p_u))$以一條線段連接到新建立節點$(p_v, \alpha(p_v), \mathrm{IRS}(p_v))$，並於線段旁標示 μ，且於新建立之節點旁標示「達成節點」，即表示該條由 p_u 到 p_v 之路徑完成。

步驟三：若無非終端節點存在，執行步驟四，否則跳至步驟二。

步驟四：繼續依步驟二～三建立另一條由起始節點至達成節點之生長樹，直到不再存有任何達成節點時才停止生長樹之建立。

步驟五：依步驟一～四，由起始節點至達成節點所建立之每一條路徑，皆可稱為推論路徑，再由這些路徑中選擇其達成節點模糊數值最大者，為最佳建議路徑。Q 表所有推論路徑之集合，S_1～S_n 表所有路徑之達成節點模糊數值，且其最佳建議路徑之模糊生產規則(FPR)如公式(4)所示，表示狀態節點 p_g 對應之狀態 $\beta(p_g) = d_g$，可信任的程度值為 z。

$Q = \{(p_g, S_1, \mathrm{IRS}(p_g)), (p_g, S_2, \mathrm{IRS}(p_g)), \ldots\ldots, (p_g, S_n, \mathrm{IRS}(p_g))\}$

$\text{Set } z = \max\{S_1, S_2, \ldots\ldots, S_n\}$

肆、資料分析與結果

在此章節中將逐一介紹本研究的實驗步驟、系統設計架構、訓練資料及測試資料的來源說明。

一、資料樣本

本研究之病毒樣本以郵件病毒為主，病毒樣本收集時間從 1999 年 04 月至 2004 年 6 月(病毒名稱根據賽門鐵克與趨勢科技網站命名方式)。在這個時間內，凡是以電子郵件作為傳播媒介的病毒(Viruses)或是網蟲(Worm)都是我們分析的對象，敘述如表 3 所示。

表 3：郵件病毒樣本

Type	Name (by Trend Corp.)
Macro	MELISSA.A，GORUM.A
Executable File	ZAUSHKA.A-O，JERM.A，COBBES.A，MAGISTR.B，KAMIL.B，BRID.A，BAGLE.P，BAGLE.Q，DUMARU.A，ZAFI.B，YOUGDOS.A
Trojan	PTWEAK.A，TROODON.A，HYBRIS.C，SIRCAM.A，GIFT.B，FEVER.A，XTC.A

C.E.P.S.

Script	JavaScript：GERMINAL.A，EXITW.A，SEEKER.A6，ACTPA.A，EXCEPTION.GEN，VBScript：REPAH.A，LOVELETTER，HARD.A，NOONER.A，INFO.A，LIFELESS.A，NEWLOVE.A，CHU.A，KALAMAR.A，CHICK.C，CHICK.B，CHICK.E，HAPTIME.B，EDNAV.B，GOOFFY.A，HEATH.A，HORTY.A，ARIC.A，VIERKA.B，GORUM.B，ZIRKO.A
Worm	NIMDA.A-O，ALIZ，PETIK.C，PETIK.A，BADTRANS.B，GIZER.A，ENVIAR.B，GOKAR.A，RADIX.A，UPDATR.A，KLEZ.E，KLEZ.H，LASTWORD.A，LOHACK.A，ZAFI.D，MERKUR.A，MYLIFE.E，PLAGE.A，PROLIN.A，SOLVINA.B，SHOHO.GEN，DESOR.A，PET.TICK.Q，PETIK.E，ZHANGPO.A，ZOHER.A，SOBIG.A，YAHA.E，BUGBEAR.A，BLEBLA.C，GAGGLE.C，SIRCAM.A，YAHA.G，LOVGATE.A，LOVGATE.B，LOVGATE.C，LOVGATE.G，COD，BAGLE.A，BAGLE.C，BAGLE.J，BAGLE.U，BAGLE.X，BAGLE.Z，NETSKY.C，NETSKY.D，MYDOOM.A，MIMAIL.A，SWEN.A，SOBER.A

二、郵件病毒行為知識本體論(Ontology)之建立

經由相關文獻探討可知，本體論 Ontology 能將知識做適當的呈現與表示，本研究根據上一節對郵件病毒行為的分析，透過 W3C 定義的 OWL 語言以及 Ontology 工具(Protégé 2000)，建立郵件病毒行為的本體論如圖 9。本研究主要強調從規格化(specification)轉換到概念化(conceptualize)的模式，希望能以本體論對郵件病毒行為作深入的探討與分析，了解其行為模式與特徵，進而有效運用本體論知識的結構，過濾郵件病毒。本研究利用 W3C 定義的相關 OWL 語法與 Ontology 操作工具(Protégé 2000)其中相關的概念敘述如下：(1) Class:類別是一個包含許多子概念集合的概念集合，此概念集合的領域涵義廣泛，蘊含許多相關的字概念集合，透過這些子集合或是樣式來敘述與呈現。(2) Property:屬性是用來敘述與呈現概念或是樣式之間的關係，可以敘述類別與類別、類別與樣式、樣式與樣式之間的關係。(3) Individual :樣式是一個體的呈現，它是某一類別的例子，用以敘述該類別所包含的概念。我們整理 12 種行為特徵，作為偵測電子郵件為正常郵件或病毒郵件的依據，如表 4。

表 4：郵件病毒特徵

變數 NO.	特徵	內容說明
X_1	附加檔案形式	Exe, vbs, scr, pif, bat, chm, com…等
X_2	附加檔案大小	附加檔案數量
X_3	附加檔案數量	附加檔案數量
X_4	內含 Script 程式	JavaScript and VBScript
X_5	內含 URL 位址	ActiveX controls URL
X_6	附加檔案	以附加檔案方式入侵

X_7	異常的 MIME 格式	異常 MIME 的檔案格式入侵
X_8	Embedded	以鑲嵌方式入侵
X_9	變化附檔之副檔名	改變副檔名，降低使用者警戒心
X_{10}	偽裝傳送者名字	採用使用者信任的名稱、暱稱等
X_{11}	偽裝主題與信件內容	以主題搭配內容欺騙使用者
X_{12}	單一收件者	單一收件者或多位收件者

（一）Ontology 之建立

本研究採用 Ontology 概念，針對郵件病毒的知識進行本體論之工程分析，並採用史丹佛大學 SMI（Standford medical informatics）研究所研發的 Protégé 2000 作為建置知識本體開發工具，並根據 Uschold and King(1995)的方法論來建置郵件病毒行為的郵件病毒知識本體，步驟如下：

1. 在郵件病毒中，我們透過之前分析的方式，來分析郵件本文並且找出郵件之所有概念，如附檔、檔案格式、附檔大小、欺瞞等方式，再將這一些概念歸納為類別或屬性，並且使用有義意的文字來表示這一些概念，概念定義如表 4。
2. 我們採用由上往下的策略來建置 ontology，先由概念中定義特定的類別及子類別，以及概念之間的關係，再依概念的共同點給予一個一般化的父類別名稱，並且把相同概念的群組在一起，產生此領域的一般性概念(General concept)。
3. OWL(Web Ontology Language)主要是以高階層的描述，進而將資源、資料儲存、與處理流程，「語意化的」關連在一起，因此我們使用 OWL 語言來表示郵件病毒的一些相關知識。
4. 找出郵件病毒的相關特徵，並且透過 OWL 描述郵件病毒的相關特徵之後，我們使用 Protégé-2000 來建造郵件病毒之間的特徵之關聯，關聯結果如圖 9。

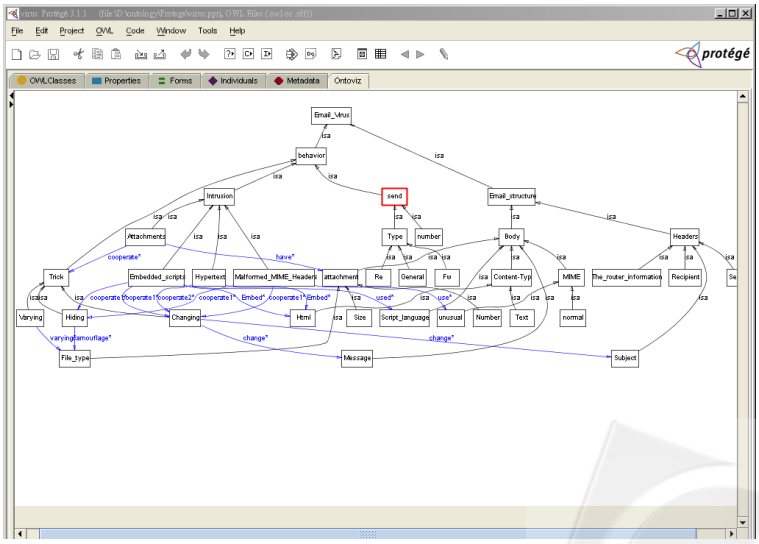


圖 9：郵件病毒行為本體論

（二）郵件病毒推論引擎之建置

經由 Uschold 和 King(1995)的本體論建構步驟，我們可得到郵件病毒的知識本體。本節將透過轉換知識本體中概念與概念之間的關係屬性所形成模糊生產規則(FPR)，而得到郵件病毒之推論派翠網路結構，並敘述推論引擎如何判斷電子郵件是否為病毒郵件。根據郵件病毒知識本體所轉換之模糊派翠網路結構(FPN)和模糊生產規則(FPR)，如圖 10 和圖 11。我們將電子郵件的 12 種郵件特徵[$x_1,x_2,\ldots,x_{12}$]編碼成[0,1,…,1]形成推論引擎的輸入型式，假設某一封電子郵件的編碼如*[1,1,1,0,0,1,0,0,0,0,1,1]*，其推論過程如下：

步驟 1： 根據模糊派翠網路結構(FPN)，該封電子信件的編碼符合 R1、R2、R3、R6、R11、R12 之規則，而起點為 X_1、X_2、X_3。

步驟 2： 根據模糊生產規則(FPR)形式(4)的推論，X_6 節點的推測值為 0.949，推論過程如下：

X_6=MAX($X_1\rightarrow X_6$，$X_2\rightarrow X_6$，$X_3\rightarrow X_6$)=MAX(1*0.898，1*0.924，1*0.949)=0.949

步驟 3： 根據模糊生產規則(FPR)形式(5)的推論，由於 X_9 與 X_{10} 節點的屬性編碼為 0，只有節點 X_{11} 滿足條件而被觸發，因此 X_9 與 X_{10} 皆無推測值，X_{11} 的推測值為，推論過程如下：

X_{11} = 0.949*0.898 =0.836

步驟 4： 根據模糊生產規則(FPR)形式(1)的推論，X_{12} 之推論值為 0.836，推論過程如下：

X_{12}=0836*1=0.836

步驟 5：節點 X_{13}：的推論過程與節點 X_{12} 相同，故推論值亦為 0.836

從節點 X_{13} 的推論值，我們可以得知這是一封高風險的病毒信件，有 83.6%的可能性為病毒郵件，並通知使用者，這封信件疑似病毒信件，請使用者選擇對此信件處理動作。

R1: IF d_1 *or* d_2 *or* d_3 THEN d_6	R9: IF d_7 THEN d_{12}
R2: IF d_4 *or* d_5 THEN d_8	R10: IF d_8 THEN d_{10}
R3: IF d_6 THEN d_9	R11 :IF d_8 THEN d_{11}
R4: IF d_6 THEN d_{10}	R12: IF d_8 THEN d_{12}
R5: IF d_6 THEN d_{11}	R13: IF d_9 THEN d_{12}
R6: IF d_6 THEN d_{12}	R14: IF d_{11} THEN d_{12}
R7: IF d_7 THEN d_{10}	R15: IF d_{10} THEN d_{12}
R8: IF d_7 THEN d_{11}	R16: IF d_{12} THEN d_{13}

圖 10：模糊生產規則(FPR)

C.E.P.S.

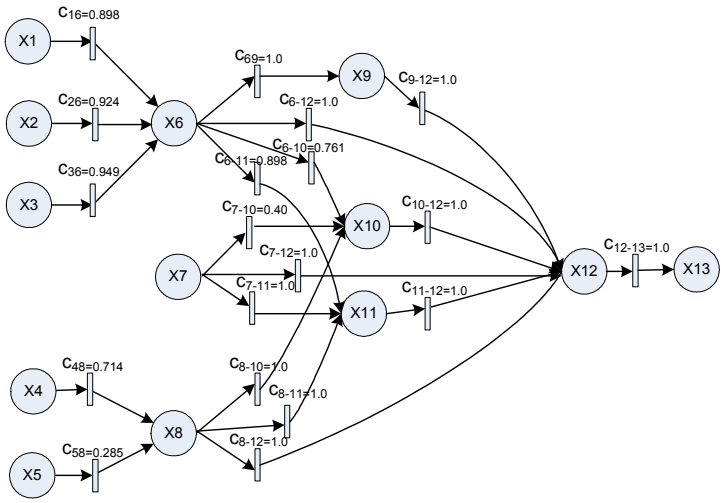


圖 11：郵件病毒行為特徵之模糊派翠網路

三、郵件病毒過濾評估

本系統之病毒過濾模組使用 1999 年到 2004 年的病毒樣本做為訓練資料，訓練出郵件病毒推論模組。為評估此一 FPN 推論模組的偵測能力，本研究以 1859 封正常郵件與訓練資料的 96 隻郵件病毒做為測試樣本，與其他資料探勘方法；SOM、Naïve Bayes 和 Decision tree 以及市面上商業的防毒軟體 Pccllin2007 和 Norton2007(病毒碼更新到 2007 四月)做比較，比較結果如表 5。

為了測試對於未知、新病毒的偵測能力，我們針對 2004 年 12 月之後的郵件病毒做測試，資料結果顯示，我們的分析模組可以偵測到新的，未知的病毒，有效解決目前商業防毒軟體最令人詬病的問題：沒有更新病毒定義檔就偵測不到病毒。表 6 顯示我們提出的方法與其他資料探勘方法；SOM、Naïve Bayes 和 Decision tree 對未知的病毒的偵測結果比較(Shih et al., 2004；2005)。我們測試之商業防毒軟體為 2007 年之版本，由於未有收集到 2007 年 4 月之後的新病毒，因此沒有將商業防毒軟體偵測的結果做呈現，但根據許多學者的研究，防毒軟體是存在不能偵測新的、未知的病毒。

表 5：偵測方法之偵測能力比較表

	TP	TN	FP	FN	Detection Rate	False Positive Rate	Overall Accuracy
Naïve Baye	89	1842	17	7	92.708%	0.914%	98.772%
Decision Tree	89	1854	5	7	92.708%	0.269%	99.642%
SOM	87	1855	4	9	90.625%	0.215%	99.335%
Pccllin2007	95	1859	0	1	98.958%	0%	99.949%
Norton2007	93	1859	0	3	96.875%	0%	99.847%
FPN	95	1855	4	1	98.958%	0.161%	99.795%

(Detection Rate：TP/ TP + FN, False Positive Rate：FP/ TN + FP, Overall Accuracy：TP + TN/ TP + TN + FN + FN)

表 6：未知病毒測試比較表（“√”= detected)

Virus Profile	FPN	SOM	Naive Bayes	Decision tree
WORM_NETSKY.P	√	√	√	√
WORM_MYTOB.A	√	√	√	√
WORM_MYTOB.K	√	-	√	√
WORM_MYTOB.R	√	√	√	√
WORM_MYDOOM.O	√	√	√	√
WORM_NETSKY.Q	√	√	√	√
WORM_SOBER.U	√	-	-	-
WORM_MYTOB.KQ	√	√	√	√
WORM_MYTOB.LL	√	√	√	
WORM_LOOKSKY.A	√	√	√	√
WORM_RONTOKBRO.C	√	√	√	√
WORM_RONTOKBRO.J	√	√	√	√
TROJ_YABE.A	√	√	√	√
TROJ_YABE.B	√	√	√	√
TROJ_BAGLE.CZ	√	-	√	√
WORM_AHKER.J	√	-	√	√
VBS_SOBER.AA	√	√	√	√
JS_WONKA.A	√	-	-	-

伍、系統架構

本研究提出一郵件病毒推論引擎，置於嵌入式系統中，目的在於防止郵件病毒入侵使用者系統。嵌入式郵件病毒過濾器架構如圖 12 所示，當使用者收取信件時，將正常與異常的電子郵件經由學習模組學習形成規則庫，並透過此規則庫來判斷是否為病毒信件。病毒過濾模組根據學習好的規則庫來判斷電子郵件是否有異常的行為，若發現電子郵件具有異常的行為時，會警告使用者，並給刪除或另存到新資料夾的建議，若電子郵件判斷為正常行為，則直接由使用者瀏覽郵件。

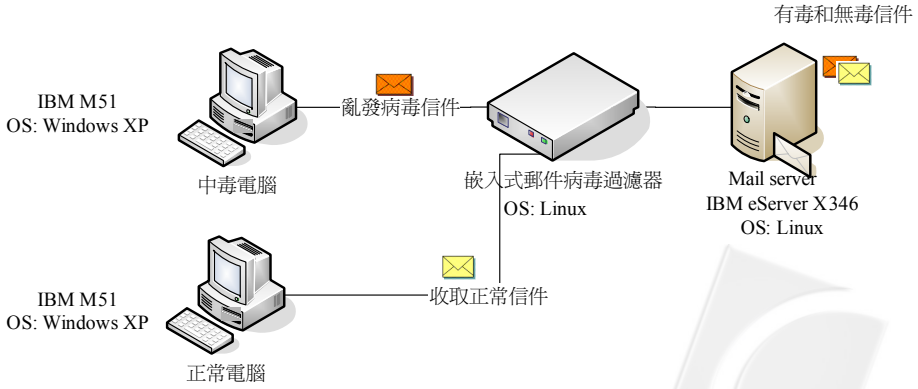


圖 12：嵌入式郵件病毒過濾器架構

一、系統開發環境

在嵌入式系統開發上，NET-Start-IXP420 是以 Intel Xscale 嵌入式產品中，專為網路相關產品開發之 IXP 420 CPU 為硬體核心，IPX420 是一個高效能的 RISC 網路應用處理器，它被廣泛被用在網路設備中，加上由晶慧資訊自行修改的最新版本 Embedded Linux 為軟體核心，如圖 13。本研究實際上所使用的軟硬體設備整理如表 7，如 WebMail 所使用的嵌入式系統、病毒專家及業務經理所使用的安全管理伺服器、收發信件的 Mail Server 及一般使用者使用的電腦等。

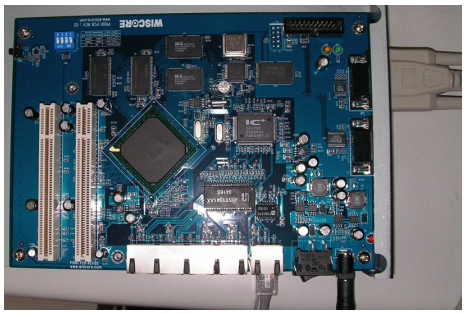


圖 13：嵌入式系統之開發板

表 7：軟硬體需求表

硬體	用途	軟體需求
嵌入式系統(EEVF)	嵌入式郵件病毒過濾器之系統	Linux–BusyBox
		Apache+PHP+IMAP
Security Server	管理買方資料及序號產生	Linux–Fedora4
		Apache+PHP+MySql+SSL
Mail Server	郵件伺服器	Linux-Fedora4
		Sendmail+IMAP
PC(3 台)	一般使用者收取信件及管理者設定	Windows–WinXP
		Browser

二、系統功能介紹

本研究之嵌入式郵件病毒過濾系統已建置完成，並已在本實驗室的網路環境下實際執行郵件病毒過濾的機制。本節將敘述本系統執行之環境設定、系統設定與操作過程說明。以下我們首先針對註冊模組與使用者驗證模組來進行說明與實作。圖 14 為實作註冊模組與使用者驗證模組之畫面，第一個步驟是管理者登入設定畫面，會要求管理者輸入帳號及密碼，若是第一次登入，會提示管理者必需到註冊頁面取得密碼。第二步驟，管理者需填入個人資料以成為我們的會員，並擁有病毒規則的更新權限。

圖 14：管理註冊模組之實作畫面圖

接著我們再針對第二個重要模組；病毒郵件過濾模組進行說明與實作，如圖 15 所示。當使用者收取電子信件時，病毒郵件過濾模組會掃描每封電子郵件，並判斷其成為病毒郵件的可能性，若某封電子信件被偵測出疑似病毒郵件，則為會出現提示畫面告知使用者，此封郵件疑似為病毒郵件，並提供 3 種功能選項讓使用者選擇所要採取的動作，我們提供的選擇有移到其它資料夾、刪除或者是瀏覽該封信件的內容。

圖 15：病毒郵件過濾模組之實作畫面圖

最後要介紹的重要模組為病毒規則更新模組，如圖 16，我們的病毒郵件過濾模組無須頻繁更新病毒規則，便能偵測新的、未知的郵件病毒，保障使用者在未更新病毒碼的狀況下，仍能保障系統安全。但若當有新病毒規則產生，表示病毒已經衍生出新的行為模式，若是沒有更新病毒規則，雖然仍有偵測病毒之能力，但病毒偵測能力仍較更新後稍弱，因此提供病毒規則更新的功能，以持續維持系統安全性。但本研究之病毒規則更新並不頻繁，相較一般商業防毒軟體頻繁的更新病毒碼，不易造成使用者困擾。

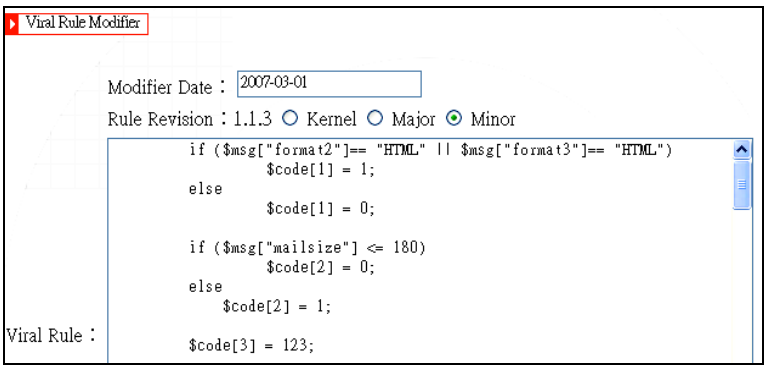


圖 16：病毒規則更新模組之實作畫面圖

陸、結論

根據文獻探討與整理，我們可以了解到新的郵件病毒不斷的產生，病毒與防毒技術不斷的互相較勁，從新病毒產生到對應的病毒碼產生仍需要一段時間，故能夠偵測未知病毒的技術一直是防毒團隊所致力研究的重點。本研究設計一套嵌入式郵件病毒過濾器，能夠偵測出已知與未知的郵件病毒，透過對郵件病毒行為的分析，找出郵件病毒最根本的本質及無法改變的行為特徵，採用本體論(ontology)建構郵件病毒之知識本體、運用模糊派翠網路(FPN)進行推論，有效防堵郵件病毒的入侵。

本研究提出之嵌入式郵件病毒過濾器，目的在於防止郵件病毒入侵使用者系統、不做侵入式安裝，透過此軟體，可以快速的設定防護機制，並且杜絕病毒信件。根據上述，本研究具有下列優點：

1. 不做侵入式安裝：目前業界防毒軟體大都採用侵入式安裝，導致系統資源佔用極大，造成使用者使用上的不便，本研究採用外接式透過嵌入式系統作為過濾中介，病毒並非進入使用者主機才被偵測過濾，而是被過濾在郵件系統之前，進一步提升系統安全性，也不會佔用或分享郵件系統資源。
2. 自動更新及未知病毒的偵測：一般防毒軟體偵測病毒是透過病毒碼比對的方式，對於每天新出現的病毒，若是沒有在病毒碼中，則是無法偵測的到，因此必需要時常的更新病毒碼。而我們的偵測方法是透過郵件病毒特徵所建立的規則庫，透過病毒特徵比對的方式可以偵測出未知的病毒，因此我們的郵件規則不必時常的更新。
3. 安全性：最有效預防郵件病毒策略為在病毒進入電腦之前就提前防護，不讓其進入電腦內部，因此我們設計之嵌入式郵件病毒系統，能做為使用者及郵件伺服器之中介，有效攔截病毒郵件，使其未能進入到使用者的電腦。

在實際執行的測試過程中，由於行為偵測方法的限制，造成誤判率稍高的問題，本研究採用使用者提醒與詢問的方式，透過使用者本身來做進一步的判

斷，來解決此一問題。我們也將考慮採用合作過濾的方式(collaborative filtering)結合多種偵測方法的優點，進一步提升病毒的偵測率與降低誤判率。

柒、致謝

感謝三位審查委員給予寶貴的意見，感謝國家科學委員會對本研究的支持與贊助，本論文為國科會補助專題研究(編號：NSC 95-3113-P-224-004)之部份成果。

參考文獻

1. 姜琇森、民國 92 年，電子郵件病毒偵測之研究，國立雲林科技大學資訊管理研究所碩士論文，。
2. 劉順德、民國 90 年，以樹狀關聯式架構偵測電子郵件病毒之探討，國立中央大學資訊管理研究所碩士論文，。
3. 黃崇益、民國 91 年，建構健保藥品給付規定本體論知識庫之研究—以降血脂用藥為例，私立台北醫學大學醫學資訊研究所碩士論文，。
4. Bunge, M., “Ontology I: The Furniture of the World. Treaties on Basic Philosophy”, Vol. 3, 1977.
5. Bernaras, A., Laresogiti, I. and Corera, J. “Building and reusing ontologies for electrical network applications”, In W. Wahlster (Ed.) European Conference on Aritficial Intelligence, 1996, pp: 298-302.
6. Chandrasekaran, B. Josephson, J. R. and Benjamins, V.R. “What are Ontologies and why do we need them? ” *IEEE Intelligent Systems*, Vol. 14, No.1, 1999, pp: 20-26.
7. Chen, S., Ke, J. S. and Chang, J. “Knowledge Representation Using Fuzzy Petri Nets,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 3, 1990, pp: 311-319.
8. Cohen, F. “Current best practice against computer viruses”, Security Technology, 1991. Proceedings. 25th Annual 1991 IEEE International Carnahan Conference on, Vol.1-3, 1991, pp: 261-270.
9. Fernandez-Lopez, M., Gomez-Perez A. and Juristo, N. “METHONOLOGY: form ontological art towards ontological engineering,” *Spring Symposium on Ontological Engineering of AAAI,* 1997, pp: 33-40.
10. Fernandez-Lopez, M., Gomez-Perez, A., Sierra, J.P. and Sierra, A.P. “Building a chemical ontology using Methontology and the Ontology Design Environment”, *IEEE Intelligent Systems*, Vol.14, No.1, 1999, pp: 37-46.

11. Gomez-Perez, A., Fernandez, A. and Vicente, M. D. “Towards a method to conceptualize domain ontologies,” European Conference on Aritficial Intelligence, 1996, pp: 41-52.
12. Gruninger, M. and Fox, M.S. “Methodology for the Design and Evaluation of Ontologies,” Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal, 1995.
13. Guarino, N. “Formal ontology and information system,” 1998. available at http://www.loa-cnr.it/Papers/FOIS98.pdf
14. International Computer Security Association Lab (ICSA Lab), “ICSA Labs 10th Annual Computer Virus Prevalence Survey,” available at http://www.icsa.net/icsa/docs/html/library/whitepapers/VPS2004.pdf
15. Kienzle, M. Darrell and Matthew, C. Elder “Recent Worms：A Survey and Trends”, ACM workshop on Rapid Malcode, Washington, DC, USA, 2003, pp1-10
16. Lee, Jieh-Sheng., Hsiang, J., Tsang, Po-Hao. “A generic virus detection agent on the internet,” System Sciences, Proceedings of the Thirtieth Hawaii International Conference, Vol.4, No.7-10, 1997, pp: 210 -219.
17. Luke, J., and Harris, C.J. “The application of CMAC based intelligent agents in the detection of previously unseen computer viruses”, *Information Intelligence and Systems*, Vol.31, 1999, pp:662 -666.
18. Murata, T. “Petri nets: Properties, analysis and application,” Proceedings of the IEEE, Vol. 77, No.4, 1989, pp: 541-580.
19. Neches, R., Fikes, R. E., Finin, T., Gruber, T. R., Senator, T. and Swartout, W. R. “Enabling technology for knowledge sharing,” AI Magazine, Vol.12, No.3, 1991, pp: 36-56.
20. OWL Web Ontology Language Overview. Available online at: http://www.w3.org/2001/sw/WebOnt/, 2005
21. Phillippo, S. J. “Practical virus detection and prevention,” Viruses and their Impact on Future Computing Systems, Vol.19, 1990, pp: 2/1-2/2.
22. Shih, D.H. “Detection of New Malicious Emails Based on Self-Organizing Maps And K-Medoids Clustering,” *J. of Information Management,* Vol. 11, No. 2, 2004, pp: 211-235.
23. Shih D. H., Chiang H. S. and Yen, D. C. “Classification Methods in the Detection of New Malicious Emails,” *Information Sciences*, Vol.172, No.1-2, 2005, pp: 241-261.
24. Staab, S., Schnurr, H.-P., Studer, R., and Sure, Y. “Knowledge processes and ontologies,” *IEEE Intelligent Systems,* Special Issue on Knowledge Management, 16(1), 2001.
25. Swarout, B., Ramesh, P. Knight, K. and Russ, T. “Toward distributed use of large-scale ontlolgy,” In A. Farquhar, M. Gruninger, A. Gomez-Perez, M. Uschool &

ven der Vet P (Eds.) AAAI'97 Spring Symposium on Ontological Engineering, 1997, pp:138-148.

26. Symantec.com, Security Response. available at http://www.Symantec.com/region/tw/enterprise/article/virus_protect.html
27. Takeshi, Okanmoto and Yoshiteru, Ishida, "An Analysis of a Model of Computer Viruses Spreading via Electronic Mail", Systems and computers in Japan, Vol. 33, No. 14, 2002, pp:81-90.
28. Trend micro, Virus Encyclopedia Search. available at http://www.trendmicro.com/vinfo/virusencyclo/default.asp
29. Uschold, M. and King, M. "Towards a methodology for building ontologies," In: IJCAI95 Workshop on Basic Ontological issues in Knowledge Sharing, pp6.1-6.10, Montreal, Canada, 1995.
30. Zenkin Denis, "Fighting Against the Invisible Enemy Methods for detecting an unknown virus," *Computers & Security*, Vol.20, No4, 2001, pp: 316-321.