知識本體於惡意程式行為分析之應用

*黃獻德　　*蔡一郎　　*邱敏乘　　**李健興　　***莊宗嚴
*國家高速網路與計算中心
**國立臺南大學資訊工程學系
***國立臺南大學數位學習科技學系
臺南市、臺灣
TonTon@nchc.org.tw

摘要

傳統電腦系統最大的威脅便是病毒感染，但目前網際網路的快速發展，木馬、後門程式等威脅也相繼衍生，惡意程式所造成的資訊安全成為不可忽略的研究議題，為了提升防護的速度以及資訊交流，不少企業組織中資訊相關單位嘗試自行建置惡意程式研究知識庫，供資安人員擬定資安規則；身處 Web 2.0 及 Semantic 的世代，除了建置知識庫供資安人員參考，採用知識本體技術建置知識庫，更可以加速其資訊技術的進一步交流，以及可以解決知識庫綱要結構不同的問題。

關鍵字: 惡意程式, 行為分析, 知識本體

I. 前言

2008 年 10 月 15 日，美國喬治亞理工學院資訊安全中心(Georgia Tech Information Security Center, GTISC)，在新興網路威脅與其對策年度高峰會，公布一份經由 GTISC 所研究以及與政府單位、產業界及學術界的重要資訊安全專家們深入訪談所得的 2009 年新興網路威脅報告，報告中列出了 2009 年消費者及商業使用者必須面對的五個最具威脅及挑戰的網路安全課題：惡意軟體、殭屍網路、網路戰爭、對網路電話及行動裝置之威脅[1]。這五種新興威脅大多藉由惡意軟體來達到資料獲取之目的。目前，大多數用戶皆採用傳統的安裝防毒軟體並更新病毒碼的方式來防護，但是根據部份特徵，比對未知可執行程式，符合病毒特徵者即判定為病毒，往往因特徵的選取不當而造成誤判，面對新型及變體惡意程式，因特徵不易取得，使其早已中毒而不知；若能針對惡意程式的行為進行分析，並建置惡意程式行為知識庫，透過已知的病毒動作，來攔截開機型、檔案感染型及特洛伊之各型未知病毒，如此便不必對程式作特殊處理，不需檢查更新病毒碼，且無需事先處理，未知惡意程式皆可能偵測出來及防止其破壞，此一技術將可以提供政府機關、企業、學校以及個人做為主動預防的方法；在 Web 2.0 的世代裡，若能運用社群運算(Social Computing）以及語意網（Semantic Web）等概念，並應用將領域知識(domain knowledge)中的抽象概念，以正規化形式提供人與代理人分享方法的知識本體（Ontology）技術來建置惡意程式行為知識庫，此種以「概念」作為知識基礎建構的網絡，將提升個人及組織對惡意程式知識利用的效能，以及防堵惡意程式擴散的速度大幅提升，並加速惡意程式行為相關防範資訊的交流。

本論文架構如下：第二節探討惡意程式分析以及知識本體相關文獻，第三節說明惡意程式行為分析系統架構，第四節說明惡意程式行為知識本體，第五節說明系統運行過程，第六節為結論。

II. 文獻探討

惡意程式所衍生出來的網路安全之問題會造成機關或企業的損失，即使有安裝傳統的防毒軟體，圖 1 即為 Infonetics Research 根據相關研究所統計之惡意程式將導致花費約年營收 2.2%的損失成本估計圖，使這些問題更顯得日漸重要，並將會是 2009 年及往後網路威脅與其對策的重點[2]；因此，透過 Ontology、Social Computing 以及惡意程式行為分析等相關研究來建置知識庫更加日漸重要，以下將做進一步的分析以及探討。

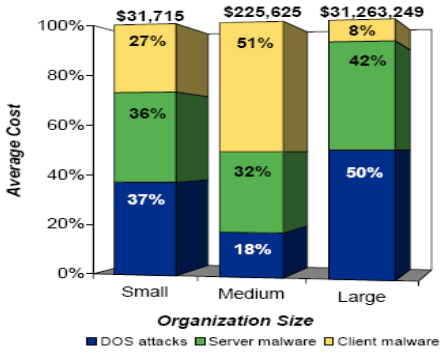

圖 1、惡意程式造成公司損失估計圖[2]

A. *惡意程式行為分析*

惡意程式的問題越來越嚴重，根據統計每天有近 250 個型式不固定的惡意程式產生[3]，傳統上，大多數公司行號透過安裝防毒軟體或是建置分析系統來自行維護；而惡意程式的分析方法可以區分為：針對惡意程式碼進行分析的程式碼分析（Code Analysis）；以及主要分析惡意程式對系統、網路等的影響運行結果之行為分析（Behavior Analysis），其中，行為分析可以從中瞭解認識惡意程式的行為特徵，從而為防毒軟體、防火牆等防禦程式的研究提供支持；表 1 即為惡意程式分析技術相關統整[4]。

表 1 惡意程式分析技術[4]

Static Analysis, Reverse Engineering (code)	Dynamic, Behavioral Analysis
● Disassembler, Debugger ● IDAPro, WinDbg, OllyDbg	● Monitor, Sandbox, VMWare ● Filemon, Regmon ● CWSandBox, Anubis, Norman
Manual Analysis	Automated Analysis
● In depth, Researcher interaction ● Requires reverse engineering skills and coding knowledge	● Fast (compression, encryption & obfuscation not a problem) and comparatively easy (usage of tolls)

惡意程式行為分析是針對惡意程式樣本是否竄改呼叫 API hooking 和 DLL injection，並且監控是否有檔案、系統登錄檔、記憶體是否被修改或創建，因此除了使用工具進行監控 Malware 的動作分析，仍需要創建完整的系統映像檔，用來跟執行 Malware 之後的系統做分析比較；動態分析比起靜態分析的缺點是一次只能執行一個惡意程式，但是靜態的分析只能根據其 Source Code 來分析，但是惡意程式函數還是可以避過這樣的分析檢查，加上越來越多的惡意程式在執行後才會透過網路下載另一真正感染破壞系統的程式；因此，針對惡意程式的行為做監控以及分析也許不是最好的方法，但是卻可以透過瞭解它的行為，並進一步的根據惡意程式的行為分類建立相關的惡意程式行為知識庫，將是目前最有效的方法[5]；另外，透過惡意程式行為分析來建置知識庫前，便需要深入瞭解惡意程式的普遍行為和方法，在分類完惡意程式的種類後，雖然有助於建置惡意程式行為知識庫，然而每天有變化多端的各種惡意程式不斷的在產生，單純建置已知的惡意程式行為知識庫，其動作較為被動，並無法達到偵測未知的相關行為是否與惡意程式行為有關聯性，且容易有語意以及綱要不同的問題；而 Semantic Web 技術中的 Ontology 即是用語意來表達物件之間的關係以及屬性，並可透過現有的知識規則進行語意推論而得到新的知識；因此，使用 Ontology 來建置惡意行為分析知識庫將會是最好的解決方法。

B. *知識本體（Ontology）*

資訊科學界的人工智慧應用研究領域廣為應用的知識本體，其最主要的功能便是「如何正規化的表達知識」[6]；目前，已經有許多學者專家針對 Ontology 定義出其架構及開發出各種應用，例如 Nguyen 學者所提出的三層式立體本體架構[7]以及 Lee 等學者所提出的應用[8-10]；圖 2 即為三層式立體領域知識本體的結構圖，其中“C_1 ,C_2 , ..., C_m ”用來表示 Concept Layer 中的每個 Concept，“I_1 ,I_2 , ..., I_m ”則表示 Instance Layer 中的每個 Instance，Relation Layer 則可以用來描述 Concept 之間的兩種關係，例如：Rc_iC_1 即表示 C_iC_1 兩者之間為 Association Relation，I_i C_i 之間則為 Instance-of Relation；圖 3 則各別表示 Concept 和 Instance 各自的結構。

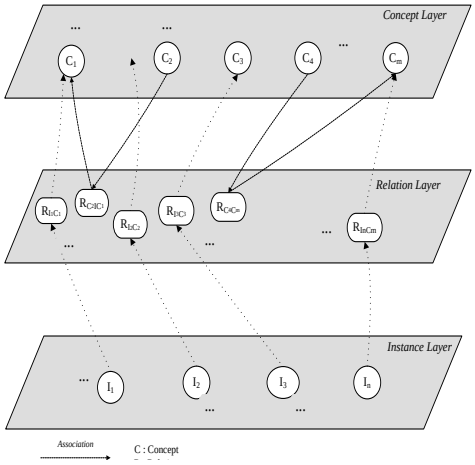

圖 2、Structure of the domain ontology[8-10]

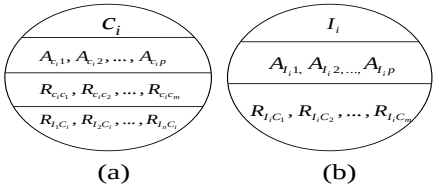

圖 3. Structure of the (a) concept (b) instance. [8-10]

由此可知，Ontology 可以透過語意清楚的描述物件與物件之間的關係；因此若採用 Ontology 來建置惡意程式行為知識庫，即便是未知程式行為，也可以透過與已知惡意程式行為之間的相互語意關係比較，進而即時阻絕其下一步侵毀行為。

III. 惡意程式行為分析系統架構

惡意程式行為分析系統平台其架構如圖 4，透過 Mwcollect、Nepenthes 以及 Honeyd 等惡意程式誘補工具，或者透過 Web2.0 之社群運算，將惡意程式樣本上傳至 Analysis Portal Server，Analysis Portal Server 會先判斷該惡意程式樣本是否曾經分析過，若未曾分析過再上傳至 Analysis Server，最後，Analysis Client 端的 PC 會將惡意程式樣本自 Analysis Server 下載至本機端運行以及監控它。

為了能快速的將感染過惡意程式的 Client 端 PC 回復到完整乾淨的狀態，以及避免惡意程式樣本偵測是否有監控工具在運行，惡意程式分析平台將採用 CWSandBox 這個 Sandbox 工具以及單純安裝的作業系統來進行惡意程式樣本的分析，並整理其產出的報告；圖 5 即為傳統 PC 在執行程式時是直接的針對 Permanent Disk Storage 做讀寫的動作，因此惡意程式便可以直接竄改系統的登錄檔、檔案以及相關設定，而 CWSandBox 是 SandBox 工具的一種，會存在於真實的系統與程式的執行中介，將所有竄改的行為予以阻隔，以達到保護系統。

除了使用 SandBox 以外，分析惡意程式樣本時還可以透過虛擬技術，如：VMWare，但是目前惡意程式的發展，不僅會攻擊竄改系統的設定檔，還可以透過 DNS 反解來驗證是否與原本欲攻擊的 IP 位址相同，來避開運行在 VMWare 所建置的虛擬機器中[4]；因此，為了分析更先進之惡意程式樣本之行為，需要建置一封閉的網路，單獨針對受感染的系統提供一般的網路服務，如：DNS、WWW、FTP 以及 IRC 等服務的 SANDNET 架構，圖 7 即為 SANDNET 概念圖；並使用 Linux 所具備的 PXE-BOOT 之功能來將被感染後的系統快速的復原以進行下一次分析[3]。

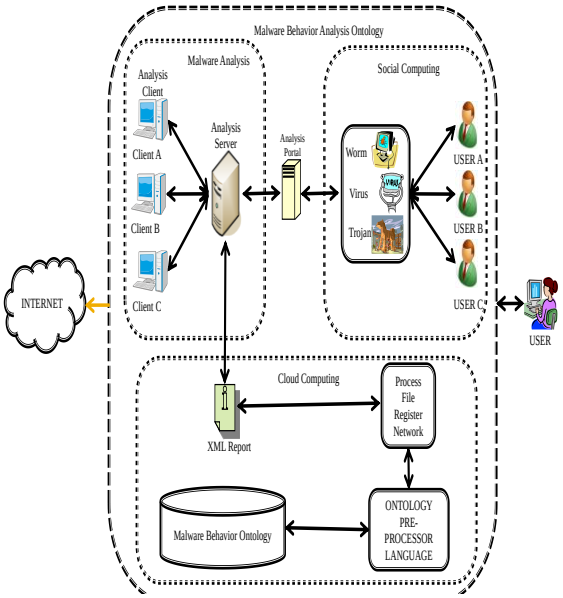

圖 4、惡意程式行為分析知識本體系統架構

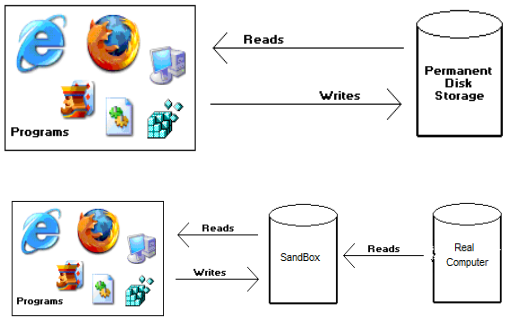

圖 5、SandBox 運行與否差異

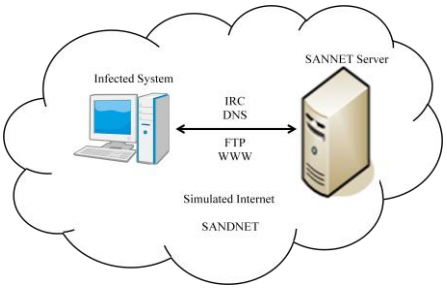

圖 6、SANDNET 概念圖[3]

IV. 惡意程式行為知識本體

表 2 為 CWSandBox 在分析完惡意程式樣本的行為後所產出的 XML 格式報告，從該報告範例中可以看出，CWSandBox 會針對惡意程式對系統所做的 dll handling、filesystem、mutex、registry、process 以及 stored created files 等行為做記錄；表 3 則是整個 XML 格式報告中關於 stored created files 的記錄；從表 3 中可以觀察出該惡意程式會自行在 C 槽建立 ab3.bat 以及 index.dat 檔，並且會將該建立出來的檔案之各相關資訊如檔案大小以及存取路徑保存起來[11]。

表 2 惡意程式行為分析 XML 範例[11]

```
<process>
<dll_handling_section>
<filesystem_section>
<mutex_section>
<registry_section>
<process_section>
<system_section>
<system_info_section>
<stored_created_files_section>
<process>
```

表 3 惡意程式行為分析 XML 範例[11]

```
<stored_created_file srcfile="C:\Documents and
Settings\Administrator\Local Settings\Temporary Internet
Files\Content.IE5\index.dat"
dstfile="e65b2507b7bee965fc1a2dcc0637adf9.dat"
filesize="32768" />
<stored_created_file srcfile="C:\Documents and
Settings\Administrator\Cookies\index.dat"
dstfile="d7a950fefd60dbaa01df2d85fefb3862.dat"
filesize="16384" />
<stored_created_file srcfile="C:\Documents and
Settings\Administrator\Local
Settings\History\History.IE5\index.dat"
dstfile="c6012e7e33fc7f7b6ff418a0e164467b.dat"
filesize="32768" />
</stored_created_files_section>
```

根據其所產出的 XML 報告，再透過三層式立體知識本體架構來繪製其概念圖，如圖 7 所示。

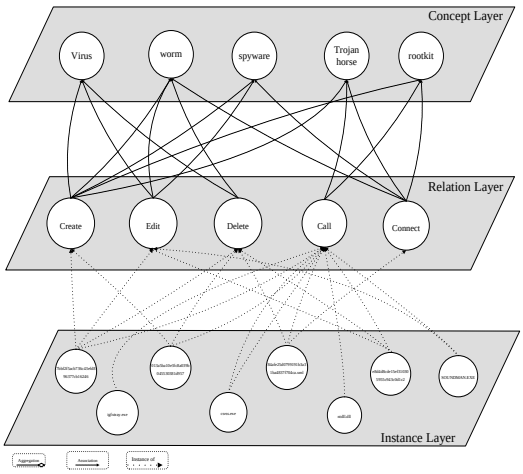

圖 7、惡意程式行為三層式立體知識本體

V. 惡意程式行為分析平臺

圖 8 即為整合了 Linux 之 PXE-BOOT 功能以及 CWSandBox 的惡意程式分析平台運行過程，其中區塊 1 為前端透過 HoneyPot 等惡意程式樣本搜集工具，區塊 2 則是分析平臺會自動將惡意程式下載至 Client 端 PC，並使用 CWSandBox 以及完整的系統執行惡意程式樣本並進行分析。

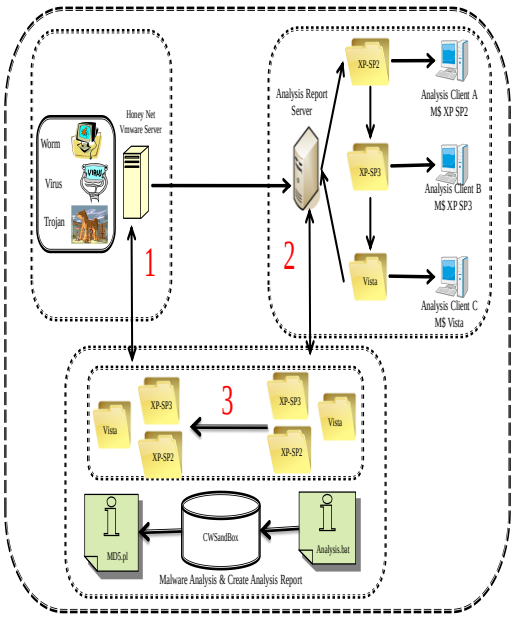

圖 8、惡意程式行為分析平台運作流程

圖 9 及圖 10 則是圖 9 中區塊 2 當系統進行分析過程時的畫面；區塊 3 是分析完惡意程式樣本後產出 XML 格式的報告，再透過 XSLT 以及 Protégé API 將其格式化成 Web 型式以及 OWL 格式，以便進行網頁呈現與 Ontology 相關後續應用推論。

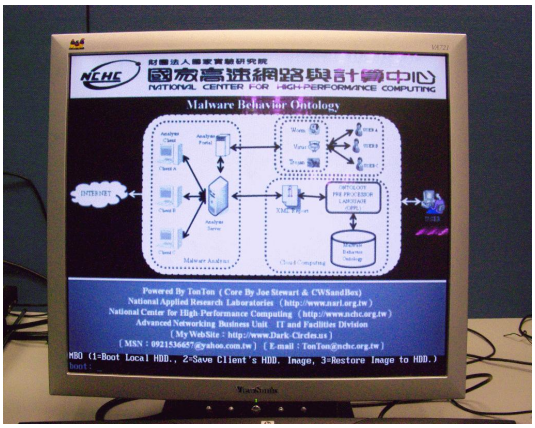

圖 9、惡意程式行為分析平台運作實際情形

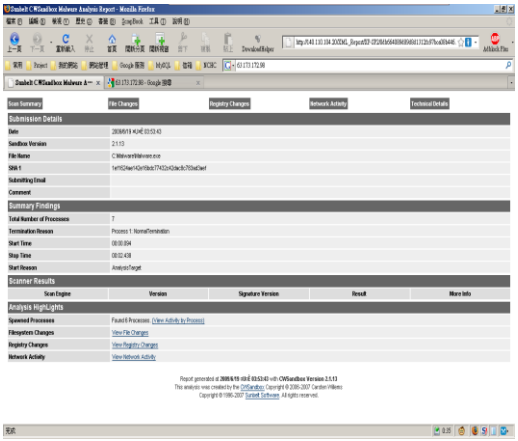

圖 12、XSLT 格式化報告[11]

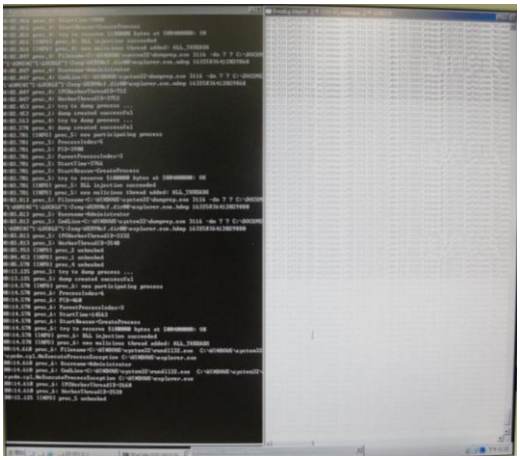

圖 10、惡意程式行為分析平臺分析過程

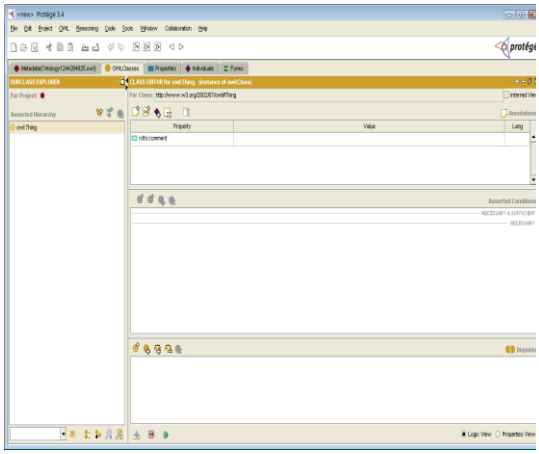

圖 13、處理後 OWL 檔匯入 Protege

圖 11 則是 CWSandBox 所產出之 XML 格式畫面，圖 12 則是 XML 檔經過 XSLT 處理所呈現的網頁畫面，圖 13 則是將已建置之 Ontology 的 OWL 檔匯入至 Protégé 的情形。

圖 14 則是當 Client 端進行完惡意程式樣本分析並產出報告上傳是 Server 端進行更進一步處理之後，Client 端會自行重新開機，並在過程中使用 PXE-BOOT 之安裝系統使其還原至原始的作業系統來進行下一次的分析。

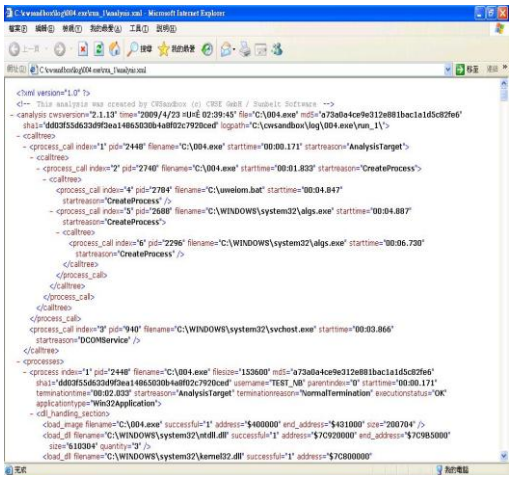

圖 11、惡意程式行為分析平台 XML 格式分析報告[11]

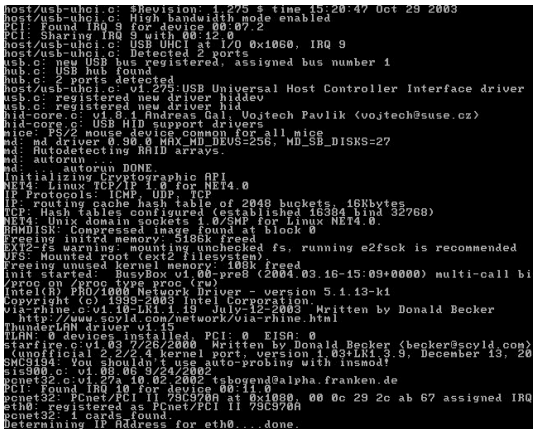

圖 14、惡意程式行為分析平台進行系統還原

VI. 結論

病毒感染是傳統電腦系統最大的威脅，加上目前網際網路快速發展，在 Web 2.0 以及 Semantic 的世代裡，更衍生出木馬、後門程式等威脅；因此，建置知識庫供資安人員參考防範將是最基本的技術，但若能將知識庫透過本論文所採用知識本體技術來建置，將可以加速其資訊技術的進一步交流，解決知識庫綱要結構問題，更可以透過知識本體針對未知之惡意程式來進行推論；另外，為了降低誤判的機率更可採用 FML（Fuzzy Makeup Language）[8-10, 12-14]表達其語意上的模糊；如此，除了解決傳統上知識表達的正規化，更可降低誤判的機率。

參考文獻

[1] G. T. I. S. Center, "Emerging Cyber Threats Report for 2009," October 15 2008.

[2] I. Research, "2008 Internet Security Outlook," 2008.

[3] Stewart Joe, "Behavioural malware analysis using Sandnets," *Computer Fraud & Security*, vol. 2006, pp. 4-6, December, 2006 2006.

[4] S. Software, "CWSandbox User Guide v 2.1.13," 2007.

[5] K. Rieck, T. Holz, C. Willems, P. Dussel, and P. Laskov, "Learning and Classification of Malware Behavior," in *Fifth Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 08)*, 2008.

[6] N. F. Noy and D. L. McGuinness, "Ontology Development 101:A Guide to Create Your First Ontology," 2003.

[7] N. T. Nguyen, "A method for ontology conflict resolution and integration on relation level," *Cybernetic and Systems*, vol. 38, pp. 781-797, 2007.

[8] C.S. Lee, M.H. Wang, and J.J. Chen, "Ontology-based Intelligent Decision Support Agent for CMMI Project Monitoring and Control," *International Journal of Approximate Reasoning*, vol. 48, pp. 62-76, 2008.

[9] C. S. Lee, M. H. Wang, Z. R. Yan, C. F. Lo, H. H. Chuang, and Y. C. Lin, "Intelligent estimation agent based on CMMI ontology for project planning," IEEE International Conference on Systems, Man, and Cybernetics (SMC 2008), Singapore, 2008.

[10] C. S. Lee, M. H. Wang, W. C. Sun, and Y. C. Chang, "Intelligent healthcare agent for food recommendation at Tainan city," IEEE International Conference on Systems, Man, and Cybernetics (SMC 2008), Singapore, 2008.

[11] C. Willems, T. Holz, and F. Freiling, "Toward Automated Dynamic Malware Analysis Using CWSandbox," IEEE Security and Privacy, vol. 5, pp. 32-39, March 2007.

[12] G. Acampora and V. Loia, "Fuzzy Control Interoperability and Scalability for Adaptive Domotic Framework," IEEE Trans. Industrial Informatics, vol. 1, pp. 97-111, May 2005.

[13] G. Acampora and V. Loia, "Using FML and Fuzzy Technology in Ambient Intelligent Environments," International Journal of Computational Intelligence Research, vol. 1, pp. 171-182, 2005.

[14] G. Acampora and V. Loia, "A Proposal of an Open Ubiquitous Fuzzy Computing System for Ambient Intelligence," Computational Intelligence for Agent-based Systems, vol. 72, pp. 1-27, 2007.