TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

**LỚP CỬ NHÂN TÀI NĂNG**

ĐỖ TRUNG ĐỨC – TRẦN QUỐC TÂM

BẢO VỆ DỮ LIỆU TRÊN

THIẾT BỊ LƯU TRỮ DI ĐỘNG

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

TP.HCM, 2015

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

**LỚP CỬ NHÂN TÀI NĂNG**

ĐỖ TRUNG ĐỨC 1112075

TRẦN QUỐC TÂM 1112276

BẢO VỆ DỮ LIỆU TRÊN

THIẾT BỊ LƯU TRỮ DI ĐỘNG

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN TIN HỌC

GIÁO VIÊN HƯỚNG DẪN

TS.TrẦn Minh TriẾt

NIÊN KHÓA 2011– 2015

|  |
| --- |
| **NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  Khóa luận đáp ứng yêu cầu của LV cử nhân tin học.  TpHCM, ngày …… tháng …… năm 2010  Giáo viên hướng dẫn |
| **NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  ………………………………………………………………………………  Khóa luận đáp ứng yêu cầu của LV cử nhân tin học.  TpHCM, ngày …… tháng …… năm 2010  Giáo viên phản biện |

LỜI CÁM ƠN

Chúng em xin chân thành cảm ơn Khoa Công Nghệ Thông Tin, trường Đại Học Khoa Học Tự Nhiên, Tp.HCM đã tạo điều kiện tốt cho chúng em thực hiện đề tài này.

Chúng em xin chân thành cảm ơn Thầy Trần Minh Triết, là người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt thời gian thực hiện đề tài. Chúng em cũng xin cảm ơn Thầy Lương Vĩ Minh, Thầy Nguyễn Đức Huy đã có những trao đổi, những chỉ dẫn giúp chúng em giải quyết các vấn đề và hoàn thiện đề tài.

Chúng em cũng xin gửi lời cảm ơn sâu sắc đến quý Thầy Cô trong Khoa đã tận tình giảng dạy, trang bị cho chúng em những kiến thức quí báu trong những năm học vừa qua.

Chúng em xin gửi lòng biết ơn sâu sắc đến Ba, Mẹ, các anh chị và bạn bè đã ủng hộ, giúp đỡ và động viên chúng em trong những lúc khó khăn cũng như trong suốt thời gian học tập và nghiên cứu.

Mặc dù chúng em đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép, nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong sự cảm thông và tận tình chỉ bảo của quý Thầy Cô và các bạn.

Nhóm thực hiện

Đỗ Trung Đức & Trần Quốc Tâm

ĐỀ CƯƠNG CHI TIẾT

|  |
| --- |
| **Tên Đề Tài**: Bảo vệ dữ liệu trên thiết bị lưu trữ di động |
| **Giáo viên hướng dẫn**: TS. Trần Minh Triết |
| **Thời gian thực hiện**: từ ngày 25/02 /2015 đến ngày 15/07 /2015 |
| **Sinh viên thực hiện:**  Đỗ Trung Đức (1112075) – Trần Quôc Tâm (1112276) |
| **Loại đề tài:** Nghiên cứu lý thuyết, giải pháp kỹ thuật và hiện thực hóa sản phẩm phần mềm |
| **Nội Dung Đề Tài** (mô tả chi tiết nội dung đề tài, yêu cầu, phương pháp thực hiện, kết quả đạt được, …):   * Nghiên cứu tổng quan tình hình thất thoát dữ liệu, đặc biệt là trên các thiết bị lưu trữ di dộng. * Phân tích các phương pháp Hook API, bảo vệ dữ liệu trước khi sao chép ra các thiết bị lưu trữ. * Phát triển ứng dụng bảo vệ dữ liệu. |

|  |  |
| --- | --- |
| Kế Hoạch Thực Hiện:  25/02/2010-01/03/2010: Tìm hiểu tình hình thất thoát dữ liệu.  01/03/2010-15/03/2010: Tìm hiểu các phương pháp phương pháp Hook API.  15/03/2010-15/04/2010: Nghiên cứu và phân tích quy trình 6 bước rút trích ontology từ dữ liệu web.  16/04/2010-29/04/2010: Đưa ra kiến trúc tổng quát cho hệ thống  02/05/2010-20/05/2010: Đưa ra các giải pháp cho các vấn đề và lựa chọn giải pháp thích hợp để xây dựng framework theo kiến trúc đã đưa ra.  21/05/2010-15/06/2010: Đưa ra các giải pháp cho các vấn đề và lựa chọn giải pháp thích hợp để hiện thực hóa các module cụ thể để sử dụng trong framework.  15/06/2010-30/06/2010: Xây dựng hoàn chỉnh framework.  01/07/2010-15/07/2010: Tiến hành chạy thử nghiệm. | |
| **Xác nhận của GVHD**  **TS. Trần Minh Triết** | **Ngày tháng năm** 2015  **Nhóm SV Thực hiện**  **Đỗ Trung Đức – Trần Quốc Tâm** |

MỤC LỤC

[LỜI CÁM ƠN 2](#_Toc416164592)

[ĐỀ CƯƠNG CHI TIẾT 2](#_Toc416164593)

[MỤC LỤC 2](#_Toc416164594)

[DANH MỤC CÁC HÌNH 2](#_Toc416164595)

[DANH MỤC CÁC BẢNG 2](#_Toc416164596)

[TÓM TẮT KHÓA LUẬN 2](#_Toc416164597)

[Chương 1 Mở đầu 2](#_Toc416164598)

[1.1. Giới thiệu 2](#_Toc416164599)

[1.2. Các hướng nghiên cứu và phát triển giải pháp bảo vệ 2](#_Toc416164600)

[1.3. Mục tiêu đề tài 2](#_Toc416164601)

[1.4. Nội dung luận văn 2](#_Toc416164602)

[Chương 2 Tổng quan về các biện pháp ngăn ngừa thất thoát dữ liệu 2](#_Toc416164603)

[2.1. McAfee 2](#_Toc416164604)

[2.1.1. Giải pháp phần mềm 2](#_Toc416164605)

[2.1.2. Giải pháp phần cứng (thường có giá thành trên $12000) 2](#_Toc416164606)

[2.1.3. Các tính năng chung của McAfee DLP Endpoint 2](#_Toc416164607)

[2.1.4. Trial McAfee Complete Endpoint Advanced 2](#_Toc416164608)

[2.2. TrendMicro 2](#_Toc416164609)

[2.2.1. Giải pháp phần mềm 2](#_Toc416164610)

[2.2.2. Các tính năng của các sản phẩm tích hợp Trend Micro Intergrated Data Loss Prevention 2](#_Toc416164611)

[2.2.3. Trial TrendMicro 2](#_Toc416164612)

[2.3. Symantec 2](#_Toc416164613)

[2.3.1. Giải pháp phần mềm 2](#_Toc416164614)

[2.3.2. Các tính năng chống mất mát dữ liệu [11] 2](#_Toc416164615)

[2.3.3. Trial Symantec File Share Encryption 2](#_Toc416164616)

[2.4. RSA 2](#_Toc416164617)

[2.4.1. Giải pháp phần mềm 2](#_Toc416164618)

[2.4.2. Các tính năng chống mất mát dữ liệu [12] 2](#_Toc416164619)

[2.5. WebSense 2](#_Toc416164620)

[2.5.1. Giải pháp phần mềm [13] 2](#_Toc416164621)

[2.6. Kết luận 2](#_Toc416164622)

[Chương 3 Can thiệp hệ thống 2](#_Toc416164623)

[3.1. Hook API 2](#_Toc416164624)

[3.2. Kỹ thuật chèn DLL vào process 2](#_Toc416164625)

[3.2.1. Registry 2](#_Toc416164626)

[3.2.2. System-wide Windows Hooks 2](#_Toc416164627)

[3.2.3. Injecting DLL by using CreateRemoteThread() API function 2](#_Toc416164628)

[3.3. Cơ chế chặn hàm API 2](#_Toc416164629)

[3.3.1. Windows subclassing 2](#_Toc416164630)

[3.3.2. Proxy DLL 2](#_Toc416164631)

[3.3.3. Code overwriting 2](#_Toc416164632)

[3.3.4. Spying by altering of the Import Address Table 2](#_Toc416164633)

[3.4. Microsoft Office Add-in 2](#_Toc416164634)

[3.4.1. Sử dụng cơ chế mã hóa mặc định của Word 2](#_Toc416164635)

[3.4.2. Tự mã hóa dữ liệu 2](#_Toc416164636)

[Chương 4 Phát hiện thiết bị lưu trữ di động tương tác với máy tính 2](#_Toc416164637)

[4.1. Giải pháp C++ 2](#_Toc416164638)

[4.2. Giải pháp C# 2](#_Toc416164639)

[4.3. Kết luận 2](#_Toc416164640)

[Chương 5 Ontology Extractor Framework rút trích ontology từ WWW **Error! Bookmark not defined.**](#_Toc416164641)

[5.1. Kiến trúc hệ thống 2](#_Toc416164642)

[5.2. Phân hệ Chuẩn bị (Preparation) 2](#_Toc416164643)

[5.2.1. Kiến trúc phân hệ 2](#_Toc416164644)

[5.2.2. Tải các trang web về lưu trữ ngoại tuyến 2](#_Toc416164645)

[5.2.3. Loại bỏ trang web không hợp lệ 2](#_Toc416164646)

[5.2.4. Chuẩn hoá các trang web 2](#_Toc416164647)

[5.3. Phân hệ Biến đổi (Transformation) 2](#_Toc416164648)

[5.3.1. Kiến trúc phân hệ 2](#_Toc416164649)

[5.3.2. Gom nhóm các trang web dựa trên đường dẫn gốc 2](#_Toc416164650)

[5.3.3. Tinh chỉnh số lượng các trang web trong một nhóm 2](#_Toc416164651)

[5.3.4. Loại trừ các thành phần lặp giữa 2 trang web 2](#_Toc416164652)

[5.3.5. Loại trừ các phần trùng nhau giữa các trang web trong cùng một nhóm. 2](#_Toc416164653)

[5.3.6. Lưu trữ các trang web vào hệ quản trị cơ sở dữ liệu 2](#_Toc416164654)

[5.4. Phân hệ Gom cụm (Instance Clustering) 2](#_Toc416164655)

[5.4.1. Kiến trúc phân hệ 2](#_Toc416164656)

[5.4.2. Lựa chọn các từ khoá 2](#_Toc416164657)

[5.4.3. Xây dựng vector đặc trưng 2](#_Toc416164658)

[5.4.4. Gom cụm các trang web dựa trên độ tương đồng giữa các vector đặc trưng 2](#_Toc416164659)

[5.5. Phân hệ Nhận diện (Recognition) 2](#_Toc416164660)

[5.5.1. Kiến trúc phân hệ 2](#_Toc416164661)

[5.5.2. Nhận diện đặc trưng cụm 2](#_Toc416164662)

[5.6. Phân hệ Liên kết (Refinement) 2](#_Toc416164663)

[5.6.1. Kiến trúc phân hệ 2](#_Toc416164664)

[5.6.2. Xây dựng đồ thị mối quan hệ giữa các khái niệm 2](#_Toc416164665)

[5.7. Phân hệ Tinh chỉnh (Revision) 2](#_Toc416164666)

[5.7.1. Kiến trúc phân hệ 2](#_Toc416164667)

[Chương 6 Kết luận 2](#_Toc416164668)

[6.1. Các kết quả đạt được 2](#_Toc416164669)

[6.2. Hướng phát triển của đề tài 2](#_Toc416164670)

DANH MỤC CÁC HÌNH

[Hình 1‑1 Số lượng thông tin rò rỉ từ năm 2006-2013 [2] 2](#_Toc416164671)

[Hình 1‑2 Tỷ lệ thông tin rò rỉ vô tình và có chủ ý năm 2012 và 2013 [2] 2](#_Toc416164672)

[Hình 1‑3 Tỷ lệ kiểu thông tin bị thất thoát [2] 2](#_Toc416164673)

[Hình 1‑4 Tỷ lệ rò rỉ dữ liệu theo các kênh [1] 2](#_Toc416164674)

[Hình 1‑5 Tỷ lệ có chủ ý và vô tình rò rỉ dữ liệu theo kênh [2] 2](#_Toc416164675)

[Hình 2‑1 Giao diện đăng nhập 2](#_Toc416164676)

[Hình 2‑2 Dashboard 2](#_Toc416164677)

[Hình 2‑3 Cấu hình chính sách bảo mật 2](#_Toc416164678)

[Hình 2‑4 Các loại file hỗ trợ 2](#_Toc416164679)

[Hình 2‑5 Giới hạn luật mã hóa 2](#_Toc416164680)

[Hình 2‑6 Device Control Setting 2](#_Toc416164681)

[Hình 2‑7 Channel Configuration 2](#_Toc416164682)

[Hình 2‑8 Action 2](#_Toc416164683)

[Hình 2‑9 Agent Management 2](#_Toc416164684)

[Hình 2‑10 Data Identifiers 2](#_Toc416164685)

[Hình 2‑11 Quản lý key 2](#_Toc416164686)

[Hình 2‑12 Lưu lịch sử hoạt động 2](#_Toc416164687)

[Hình 2‑13 Mã hóa file dạng Zip 2](#_Toc416164688)

[Hình 2‑14 Mã hó ổ đĩa 2](#_Toc416164689)

[Hình 2‑15 Xem thông tin file mã hóa 2](#_Toc416164690)

[Hình 2‑16 Chia sẻ file mã hóa 2](#_Toc416164691)

[Hình 3‑1 Hook Server can thiệp vào 2 ứng dụng [14] 2](#_Toc416164692)

[Hình 3‑2 User mode và Kernel mode trong Windows 2](#_Toc416164693)

[Hình 3‑3 PE format khi thực thi chương trình 2](#_Toc416164694)

[Hình 3‑4 Mã hóa dữ liệu của Microsoft Word 2](#_Toc416164695)

[Hình 3‑5 Giao diện mới khi lock file 2](#_Toc416164696)

[Hình 3‑6 InstallShield Limited Edition 2](#_Toc416164697)

[Hình 3‑7 InstallShield Limited Edition Download 2](#_Toc416164698)

[Hình 3‑8 InstallShield Limited Edition Error 2](#_Toc416164699)

[Hình 3‑9 Cài đặt WordAddin1 2](#_Toc416164700)

[Hình 3‑10 Cài đặt WordAddin1 2](#_Toc416164701)

[Hình 3‑11 Word Options – Trust Center 2](#_Toc416164702)

[Hình 3‑12 Trust Center – Add-ins 2](#_Toc416164703)

[Hình 3‑13 Quản lý Add-ins trong Word 2](#_Toc416164704)

[Hình 4‑1 Giao diện ban đầu 2](#_Toc416164705)

[Hình 4‑2 Chương trình thông báo khi máy tính nhận USB 2](#_Toc416164706)

[Hình 4‑3 Chương trình thông báo khi USB không tương tác 2](#_Toc416164707)

[Hình 4‑4 Chương trình thông báo khi có USB tương tác 2](#_Toc416164708)

[Hình 4‑5 Chương trình thông báo khi USB không còn tương tác 2](#_Toc416164709)

[Hình 4‑6 Thay thế một USB khác 2](#_Toc416164710)

[Hình 5‑1 Kiến trúc tổng quát Ontology Extractor Framework 2](#_Toc416164711)

[Hình 5‑2 Kiến trúc các phân hệ chính của Ontology Extractor Framework 2](#_Toc416164712)

[Hình 5‑3 Giao diện Phân hệ Chuẩn bị 2](#_Toc416164713)

[Hình 5‑4 Kiến trúc Phân hệ Chuẩn bị 2](#_Toc416164714)

[Hình 5‑5 Mô hình xử lý tổng quát Phân hệ Chuẩn bị 2](#_Toc416164715)

[Hình 5‑6 Giao diện phân hệ Biến đổi 2](#_Toc416164716)

[Hình 5‑7 Kiến trúc Phân hệ Biến đổi 2](#_Toc416164717)

[Hình 5‑8 Mô hình xử lý tổng quát Phân hệ Biến đổi 2](#_Toc416164718)

[Hình 5‑9 Các đường dẫn gốc được phân nhóm 2](#_Toc416164719)

[Hình 5‑10 Cây đường dẫn 2](#_Toc416164720)

[Hình 5‑11 So sánh và đánh dấu thành phần lặp giữa 2 cây cấu trúc HTML 2](#_Toc416164721)

[Hình 5‑12 Tính độ tương đồng cây cấu trúc HTML 2](#_Toc416164722)

[Hình 5‑13 Xpath 2](#_Toc416164723)

[Hình 5‑14 Giao diện phân hệ Gom cụm 2](#_Toc416164724)

[Hình 5‑15 Kiến trúc Phân hệ Gom cụm 2](#_Toc416164725)

[Hình 5‑16 Mô hình xử lý tổng quát Phân hệ Gom cụm 2](#_Toc416164726)

[Hình 5‑17 Giao diện phân hệ Nhận diện 2](#_Toc416164727)

[Hình 5‑18 Kiến trúc Phân hệ Nhận diện 2](#_Toc416164728)

[Hình 5‑19 Mô hình xử lý tổng quát Phân hệ Nhận diện 2](#_Toc416164729)

[Hình 5‑20 Giao diện phân hệ Liên kết 2](#_Toc416164730)

[Hình 5‑21 User Control phân hệ Liên kết 2](#_Toc416164731)

[Hình 5‑22 Mô hình xử lý tổng quát Phân hệ Liên kết 2](#_Toc416164732)

[Hình 5‑23 Giao diện phân hệ Tinh chỉnh 2](#_Toc416164733)

[Hình 5‑24 User Control phân hệ Tinh chỉnh 2](#_Toc416164734)

[Hình 5‑25 Mô hình xử lý tổng quát Phân hệ Tinh chỉnh 2](#_Toc416164735)

DANH MỤC CÁC BẢNG

[Bảng 5‑1 Các lớp chính cấu thành phân hệ Chuẩn bị 2](#_Toc416164736)

[Bảng 5‑2 Các bộ lọc trang web 2](#_Toc416164737)

[Bảng 5‑3 Các lớp chính cấu thành phân hệ Biến đổi 2](#_Toc416164738)

[Bảng 5‑4 Các lớp chính cấu thành phân hệ Gom cụm 2](#_Toc416164739)

[Bảng 5‑5 Các bộ lọc từ khóa 2](#_Toc416164740)

[Bảng 5‑6 Các lớp chính cấu thành phân hệ Nhận diện 2](#_Toc416164741)

TÓM TẮT KHÓA LUẬN

Với sự xuất hiện mạnh mẽ của web ngữ nghĩa, khái niệm ontology ngày càng được đề cập đến nhiều hơn. Nhu cầu sử dụng ontology tăng cao không chỉ trong lĩnh vực web ngữ nghĩa mà còn trong nhiều lĩnh vực khác cần có nguồn thông tin giàu ngữ nghĩa do ontology đem lại phục vụ cho nhu cầu thực hiện các suy diễn và các thuật toán một cách tốt hơn. Do nhu cầu cao như vậy nên các nghiên cứu về phương pháp xây dựng ontology nhận được nhiều sự quan tâm trong giới nghiên cứu. Những thực tế này cho thấy nhu cầu xây dựng nên các hệ thống rút trích ontology một cách bán tự động hoặc tự động trở nên rất cấp thiết. Do đó mục tiêu của khóa luận này nhằm nghiên cứu về các vấn đề liên quan đến ontology và các phương pháp rút trích nhằm xây dựng một framework để hiện thực hóa một quy trình rút trích ontology.

Nội dung của đề tài này tập trung vào việc *Nghiên cứu và phát triển hệ thống rút trích ontology từ web*. Ngoài việc nghiên cứu các vấn đề liên quan đến ontology, khóa luận còn tập trung vào phân tích quy trình rút trích ontology từ web và xây dựng framework để hiện thực hóa quy trình này. Khóa luận còn xây dựng các plugin cụ thế để sử dụng trên framework này.

Nội dung khóa luận bao gồm 6 chương:

**Chương** 1**:** Mở đầu

**Chương** 2**:** Tổng quan Ontology

**Chương** 3**:** Các hướng tiếp cận trong việc xây dựng Ontology

**Chương** 4**:** Quy trình rút trích Ontology từ WWW

**Chương** 5**:** Ontology Extractor Framework rút trích ontology từ WWW

**Chương** 6**:** Kết luận

# Mở đầu

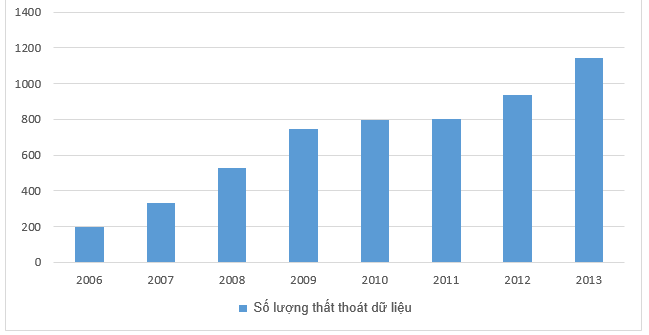
* Nội dung Chương 1 giới thiệu tổng quan về đề tài, phân tích nhu cầu thực tế và lý do thực hiện đề tài. Trên cơ sở đó, nội dung Chương 1 trình bày mục tiêu và nội dung thực hiện của đề tài.

## Giới thiệu

***Ngăn ngừa thất thoát dữ liệu*** (**D**ata **L**oss **P**revention hoặc **D**ata **L**eakage **P**revention – DLP) là một trong những vấn đề quan trọng trong lĩnh vực an toàn thông tin. Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, dữ liệu được lưu trữ và trao đổi qua nhiều phương tiện khác nhau cũng tăng vọt, kéo theo nhu cầu phát hiện và ngăn ngừa thất thoát dữ liệu ngày càng dược quan tâm.

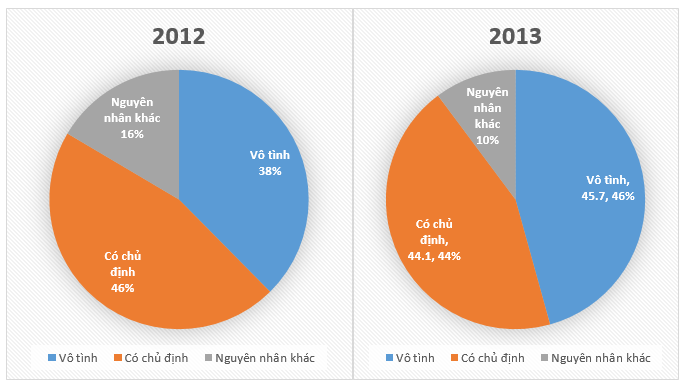
Ổ đĩa USB, máy nghe nhạc MP3, đĩa CD, DVD, và phương tiện di động có khả năng lưu trữ khác rất tiện ích, nhưng cũng đặt ra một mối đe dọa thực sự đối với cá nhân và tổ chức sử dụng chúng. Với kích thước nhỏ và dung lượng lưu trữ rất lớn làm cho các thiết bị này dễ dàng chứa **các dữ liệu mật** của khách hàng và sở hữu trí tuệ để đi thẳng ra cửa trước và rơi vào tay người xấu. Nguyên nhân có thể là do thông qua sự mất mát hoặc trộm cắp. Làm thế nào để bạn biết ai là người lưu trữ những gì trên thiết bị? Và thậm chí nếu người hoặc những người có quyền sử dụng các dữ liệu của bạn, làm thế nào bạn có thể chắc chắn họ đang giữ nó an toàn?

Báo cáo của **InfoWatch** cho thấy mức độ nghiêm trọng của DLP. InfoWatch là tổ chức thu thập và phân tích thất thoát dữ liệu hàng năm. Cụ thể, thất thoát dữ liệu qua từng năm ngày càng tăng cả về số lượng lẫn phương thức. Năm 2012, có tới **934** bản thông tin mật bị thất thoát, tăng 16% so mới năm trước đó [1]. Số lượng này tăng 22% vào năm 2013, tức là có **2143** bản thông tin mật bị mất. Nửa đầu năm 2014, số thông tin rò rỉ đã tăng 32% so với cùng kỳ năm 2013 [1] [2] [3].



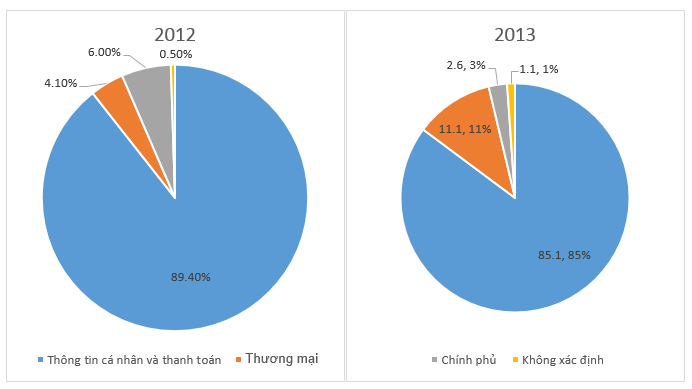
Hình ‑1 Số lượng thông tin rò rỉ từ năm 2006-2013 [2]

Thông qua báo cáo của InfoWatch, những dữ liệu thất thoát tập trung vào 3 nguyên nhân chính: Vô tình (Accidental), Có chủ định (Indentional) và các nguyên nhân khác (Unspecified) [1] [2].



Hình ‑ Tỷ lệ thông tin rò rỉ vô tình và có chủ ý năm 2012 và 2013 [2]

Trong khối lượng thông tin rò rỉ khổng lồ đó, thông tin cá nhân và thông tin thanh toán chiếm tỷ lệ nhiều nhất (gần 90%) [1] [2] [3].

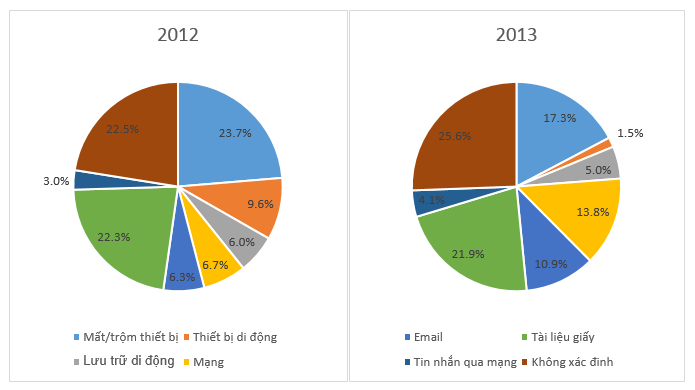


Hình ‑ Tỷ lệ kiểu thông tin bị thất thoát [2]

Những số liệu trên Hình 1‑2 và Hình 1‑3 cho thấy một phần lớn dữ liệu bị thất thoát là do sự vô ý của cá nhân.

Cuối cùng, các phương thức thất thoát dữ liệu cũng rất đa dạng. Bị đánh cắp/mất thiết bị điện tử, thiết bị di động và các thiết bị thông tin di động là những kênh rò rỉ dữ liệu liên quan đến vấn đề cần mã hóa thông tin.

Các thiết bị lưu trữ thông tin hiện nay ngày càng nhỏ gọn về kích thước nhưng dung lượng bộ nhớ lại tăng cao. Chính vì vậy, số lượng lớn dữ liệu quan trọng dễ dàng rõ rỉ ra ngoài qua các thiết bị lưu trữ di động. Thống kê dưới đây cho ta cái nhìn cụ thể hơn về vấn đề này.



Hình ‑ Tỷ lệ rò rỉ dữ liệu theo các kênh [1]

Hình 1‑4 cho thấy tỷ lệ thất thoát dữ liệu qua các thiết bị lưu trữ là không hề nhỏ. Năm 2012, có tới 23,7% là do mất/trộm thiết bị và 13,6% là do các thiết bị lưu trữ di động. Tỷ lệ rò rỉ dữ liệu theo các kênh ở hình Hình 1‑4 trên cho thấy: nếu có thể mã hóa thông tin trên các thiết bị như Laptop, USB… sẽ là một biện pháp hữu hiệu ngăn chặn thất thoát dữ liệu. Những dữ liệu này thường là do vô ý đánh mất dữ liệu như theo báo cáo năm 2013 của InfoWatch:

Hình ‑ Tỷ lệ có chủ ý và vô tình rò rỉ dữ liệu theo kênh năm 2013 [2]

Theo số liệu từ Hình 1‑5, tỷ lệ thất thoát dữ liệu trên thiết bị lưu trữ di động do vô tình cao hơn nhiều so với tỷ lệ có chủ ý. Do vậy, việc nghiên cứu và phát triển giải pháp nhằm ngăn chặn thất thoát dữ liệu do vô tình là rất cần thiết.

## Lý do thực hiện đề tài

## Mục tiêu đề tài

Nghiên cứu về các phương pháp bảo vệ dữ liệu, đặc biệt là dữ liệu trên các thiết bị lưu trữ di động. Những thiết bị này có thể bị mất/đánh cắp dễ dàng do sự vô ý của người dùng. Phát triển phần mềm tự động bảo vệ dữ liệu khi sao chép ra các thiết bị này.

## Nội dung luận văn

Luận văn bao gồm 6 chương, nội dung chính từng chương như sau:

**Chương** 1**:** Mở đầu

Trình bày sơ lược Data Leakage Prevention – DLP, tình hình thất thoát dữ liệu hiện nay. Nhu cầu bảo vệ dữ liệu trên các thiết bị lưu trữ di động.

**Chương** 2**:** Tổng quan Ontology

Trình bày sơ lược nguồn gốc của khái niệm ontology cũng như các định nghĩa khác nhau về ontology của các nhóm tác giả. Ngoài ra, nội dung chương còn đề cập đến các loại ontology khác nhau cũng như các cách biểu diễn ontology và ứng dụng cụ thể của những ontology vào các lĩnh vực khác nhau.

**Chương** 3**:** Các hướng tiếp cận trong việc xây dựng Ontology

Trình bày các nguồn dữ liệu được sử dụng trong việc xây dựng ontology cũng như tổng quan các phương pháp xây dựng nên ontology. Nội dung chương còn để cập đến tổng quát các phương pháp cụ thể để xây dựng ontology của các công trình nghiên cứu khác nhau.

**Chương** 4**:** Quy trình rút trích Ontology từ WWW

Nội dung của **Error! Reference source not found.** trình bày, tóm tắt và phân tích phương pháp rút trích onotlogy từ WWW được các tác giả Du C. Timon, Li Feng, và King Irwin đề xuất [4]. Dựa trên phương pháp này, chúng em đã nghiên cứu và đề xuất một số giải pháp nhằm hiện thực hóa và cải tiến quy trình rút trích ontology từ WWW. Những giải pháp hiện thực hóa và cải tiến này được chúng em trình bày chi tiết trong **Error! Reference source not found.**.

**Chương** 5**:** Ontology Extractor Framework rút trích ontology từ WWW

Nội dung **Error! Reference source not found.** trình bày hệ thống framework được nhóm phát triển để hiện thực hóa quy trình rút trích ontology từ WWW đã được trình bày ở **Error! Reference source not found.**. Các vấn đề và giải pháp khi xây dựng framework và hiện thực hóa quy trình cũng được trình bày trong chương này.

**Chương** 6**:** Kết luận và hướng phát triển

Nội dung của Chương 6 trình bày các kết quả đã đạt được và hướng phát triển của đề tài.

# Tổng quan về các biện pháp ngăn ngừa thất thoát dữ liệu

* Nội dung Chương 2 giới thiệu tổng quan về các biện pháp ngăn ngừa thất thoát dữ liệu của các tổ chức bảo mật uy tín. Khảo sát các phần mềm tương ứng được hỗ trợ cho người dùng bảo vệ dữ liệu

## McAfee

### Giải pháp phần mềm

Đối với các công ty kinh doanh nhỏ ( Small business) thì có các loại sản phẩm có các chức năng liên quan đến **DLP**. ( McAfee DLP Endpoint) [5]

1. Endpoint Protection Essential for SMB giá thành từ $22.10 - $24.65 trong 1 năm cho một license được thiết kế cho các công ty kinh doanh nhỏ với 250 thiết bị hoặc ít hơn.
2. Endpoint Protection Advanced for SMB giá thành từ $44.20 - $50.15 cho một license trong 1 năm tương tự như Endpoint Protection Essential nhưng được tích hợp thêm một số tính năng khác như bảo vệ giám sát các thông tin trên các thiết bị mobile như android, mã hóa các thông tin trên các thiết bị …
3. Endpoint Protection Suite giá thành từ $33.69- $48.12 cho một license trong 1 năm được sử dụng tích hợp trong các giải pháp an ninh.
4. Endpoint Protection Advanced. $69.34 - $99.07 cho một license trong 1 năm tương tự như Endpoint Protection Suite và được tích hợp thêm các chức năng bảo vệ khác.

Đối với các doanh nghiệp lớn thì có các bộ sản phẩm khác tương tự cũng tích hợp McAfee Device Control, McAfee Application Control … (**DLP** ) để bảo vệ ngăn chặn các sự cố thất thoát dữ liệu. Giá thành các sản phẩm này sẽ được thỏa thuận giữa reseller và doanh nghiệp [6]. Hai phiên bản cho các doanh nghiệp lớn:

* McAfee Complete Endpoint Protection – Enterprise
* McAfee Complete Endpoint Protection – Business

### Giải pháp phần cứng (thường có giá thành trên $12000)

McAfee DLP 4400 Copper Appliance: ứng dụng DLP 4400 là nền tảng Intel-based platform với 6 nhân CPU, 24 GB RAM và hơn 8 TB lưu trữ trong thành phần 2U.

McAfee DLP 5500 Copper Appliance: ứng dụng DLP 4400 là nền tảng Intel-based platform với 6 nhân CPU, 32 GB RAM và hơn 10 TB lưu trữ trong thành phần 2U.

DLP hardware appliance solutions are self-contained devices. There are no minimum software or hardware system requirements or additional database installations required.

McAfee DLP 4400 Copper Appliance: The DLP 4400 appliance is an Intel-based platform with dual six core CPUs, 24 GB of RAM and over 8 TB of storage in a 2U form factor. DLP hardware appliance solutions are self-contained devices. There are no minimum software or hardware system requirements or additional database installations required.

McAfee DLP 5500 Copper Appliance: The DLP 5500 appliance is an Intel-based platform with dual six core CPUs, 32 GB of RAM and over 10 TB of storage in a 2U form factor. DLP hardware appliance solutions are self-contained devices. There are no minimum software or hardware system requirements or additional database installations required.

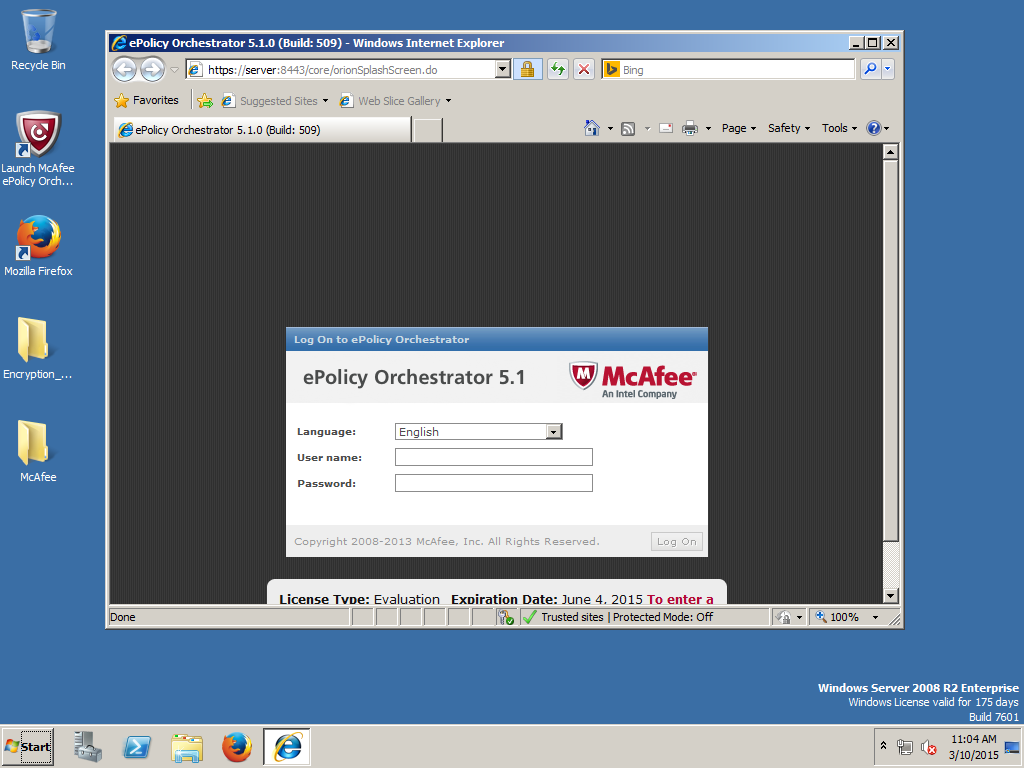
### Các tính năng chung của McAfee DLP Endpoint

**Protect and Comply**: Theo dõi các sự kiện thời gian thực và áp dụng các chính sách an ninh một cách nhanh chóng và dễ dàng để điều chỉnh và hạn chế việc nhân viên sử dụng và truyền tải dữ liệu nhạy cảm mà không ảnh hưởng đến năng suất lao động. Bảo vệ dữ liệu khỏi các mối đe dọa xuất phát từ bên trong như email, IM, các trang web, sao chép dữ liệu qua USB và công việc in ấn. Ngoài ra còn có thể ngăn chặn mất mát các dữ liệu mất khỏi các chương trình Trojans, worms và các ứng dụng chia sẻ tập tin khác.

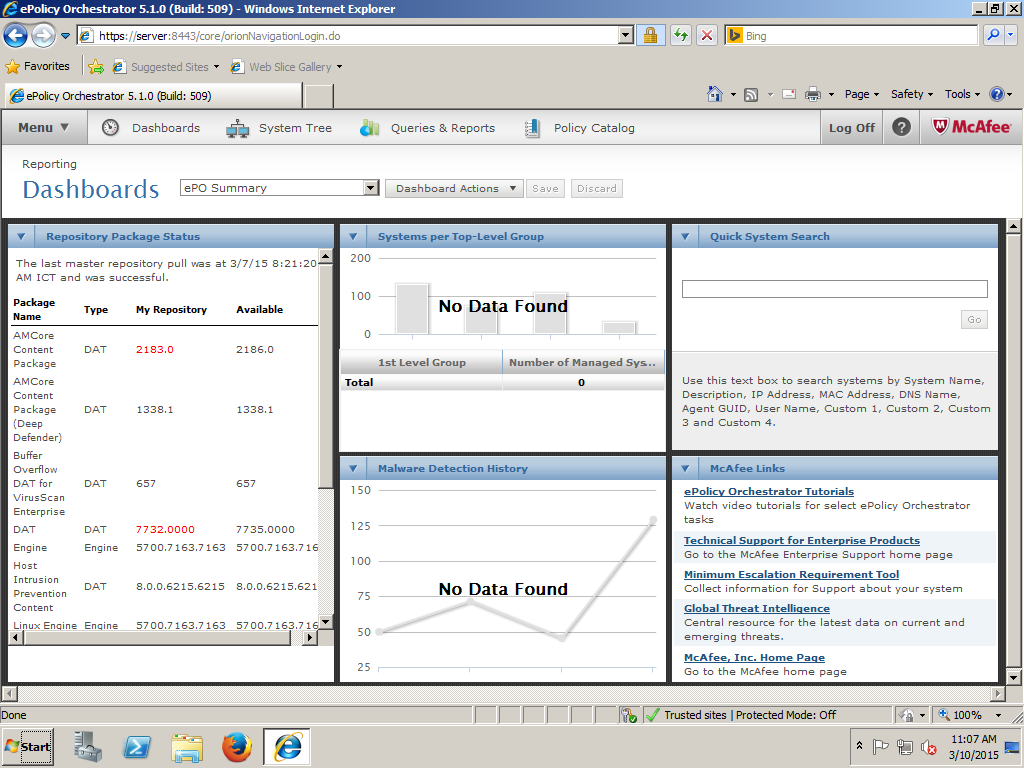
**Control and Manage**: Công cụ quản lý thiết bị toàn diện giúp cho ta kiểm soát và ngăn chặn các dữ liệu mật được sao chép vào USBs, ổ đĩa flash, CD/ DVD, iPod và các thiết bị lưu trữ di động khác. Các thông số thiết bị như product ID, vendor ID, số serial, loại thiết bị, và tên thiết bị có thể được xác định và phân loại. Hơn nữa, các chính sách khác nhau như chặn hoặc mã hóa có thể được áp dụng cho nội dung các thông tin được tải lên các thiết bị.

**Centralized Management Through McAfee ePO Software**: Tích hợp với phần mềm McAfee EPO cung cấp giám sát sự kiện thời gian thực và chính sách tập trung và quản lý sự cố. Nó cho phép dễ dàng thu thập các dữ liệu sử dụng quan trọng như người gửi, người nhận, thời gian và các chứng cứ dữ liệu. Với một cú click chuột, phần mềm McAfee cho ra các báo cáo chi tiết để chứng minh với các kiểm toán viên, quản lý cấp cao, và các bên liên quan cấp rằng các biện pháp và quy định bên trong đã được áp dụng đúng chỗ.

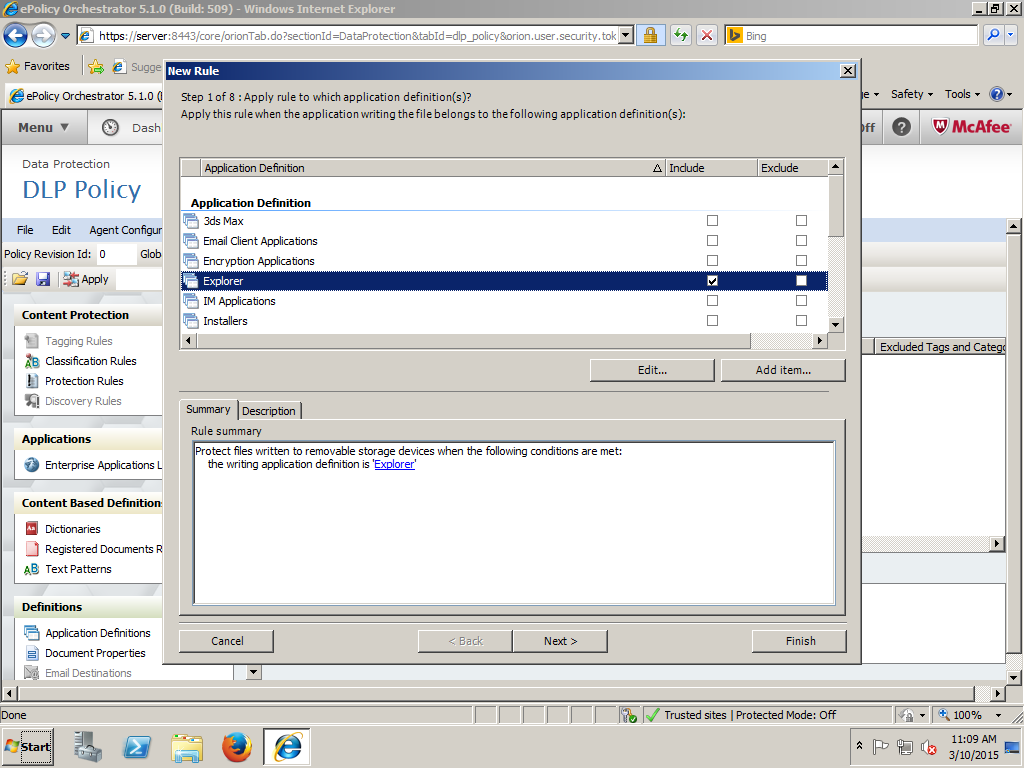
### Trial McAfee Complete Endpoint Advanced



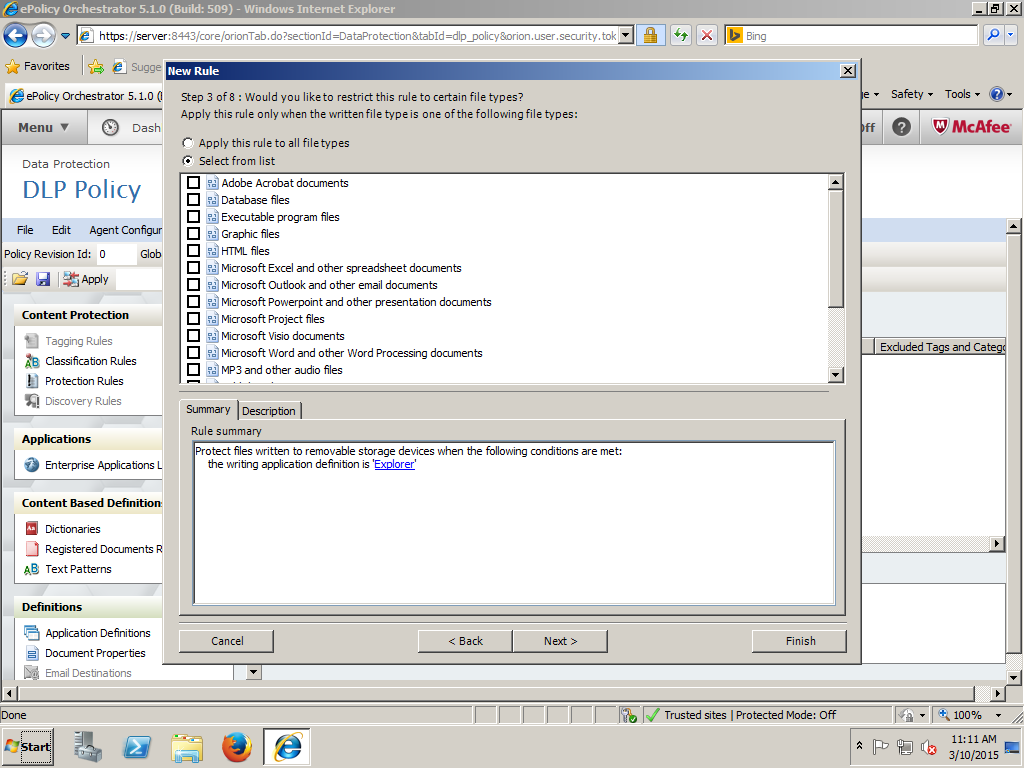
Hình ‑ Giao diện đăng nhập

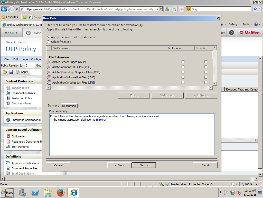


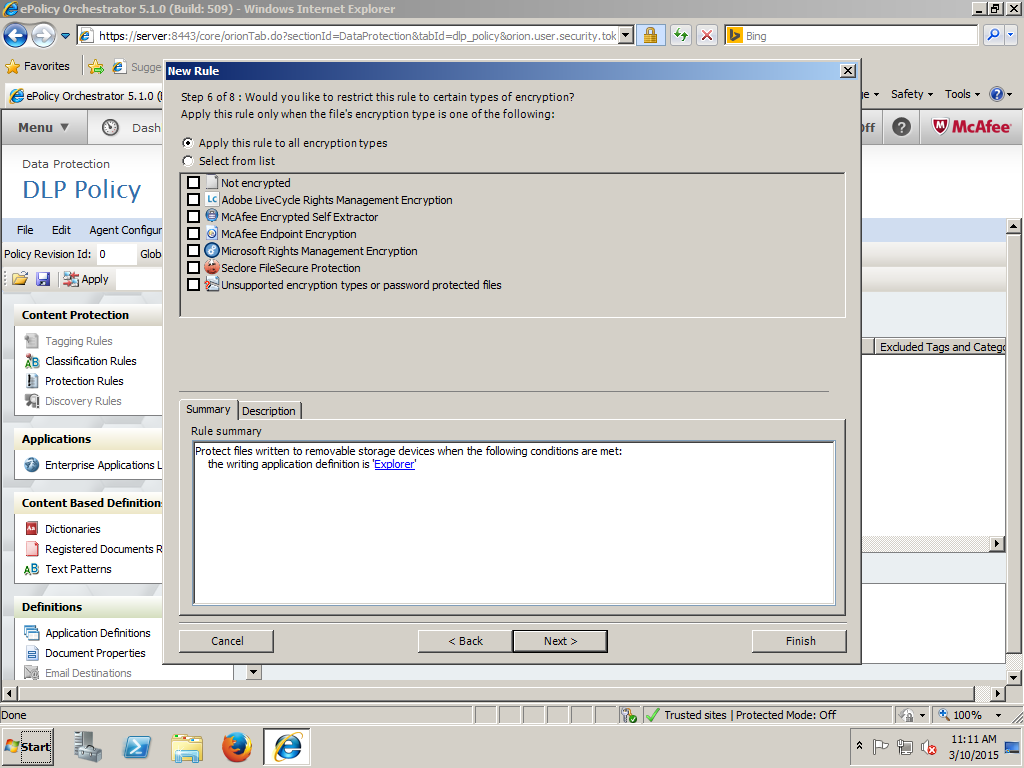
Hình ‑ Dashboard



Hình ‑ Cấu hình chính sách bảo mật



Hình ‑ Các loại file hỗ trợ



Hình ‑ Giới hạn luật mã hóa

## TrendMicro

### Giải pháp phần mềm

Các giải pháp phần mềm hiện tại hầu hết của Trend Micro đa số đều được tích hợp thêm DLP plug-in.

Đối với các SMB thì có các gói sản phẩm Worry-Free [7]:

* TrendMicro Worry-Free Business Security Services $29.06/user ( 100 user)
* Trend Micro Worry-Free™ Business Security Standard $32.36/user( 100 user)
* TrendMicro Worry-Free Business Security Advanced $59.32/ user (100 user)

Nếu như các gói này được mua với số lượng từ 2-100 thì sẽ có các giá thành khác nhau. Ví dụ như gói TrendMicro Worry-Free Business Security Advanced sẽ có giá $124.04/year với 2 người dùng còn với 100 users giá sẽ là $5932/year.

Đối với các doanh nghiệp và midsize thì có nhiều các giải pháp khác nhau với các giá thành thương lượng [8] giữa nhà cung cấp và doanh nghiệp như:

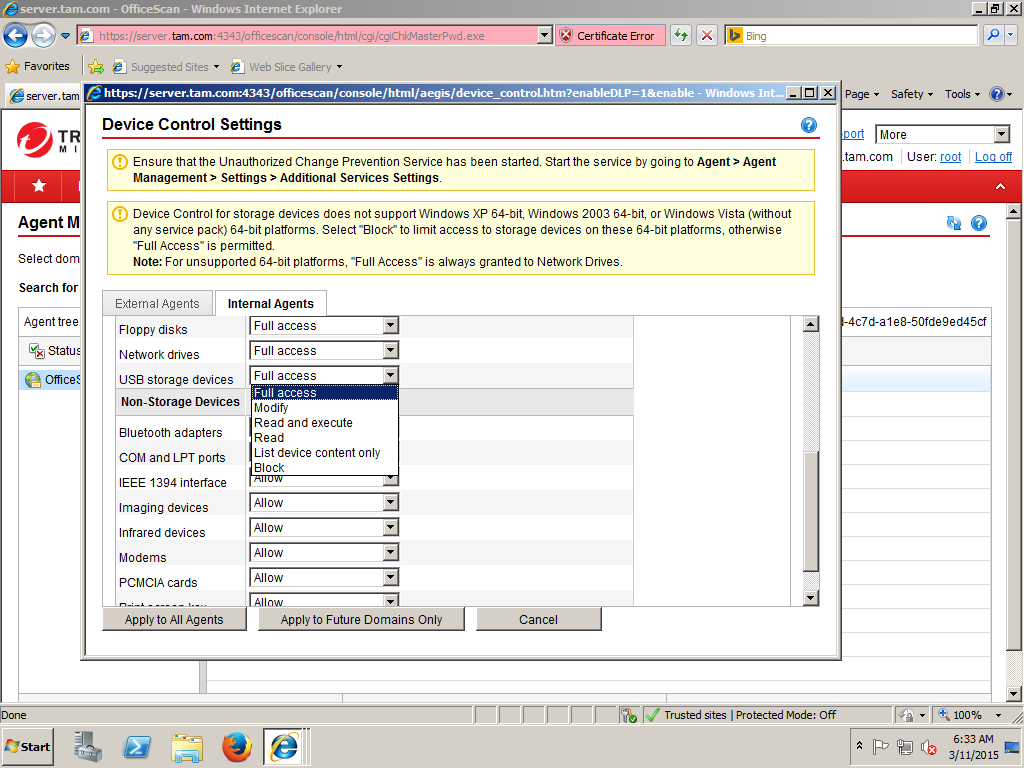
* Cloud and Data Center Security
* Complete User Security
* Custom defend

### Các tính năng của các sản phẩm tích hợp Trend Micro Intergrated Data Loss Prevention

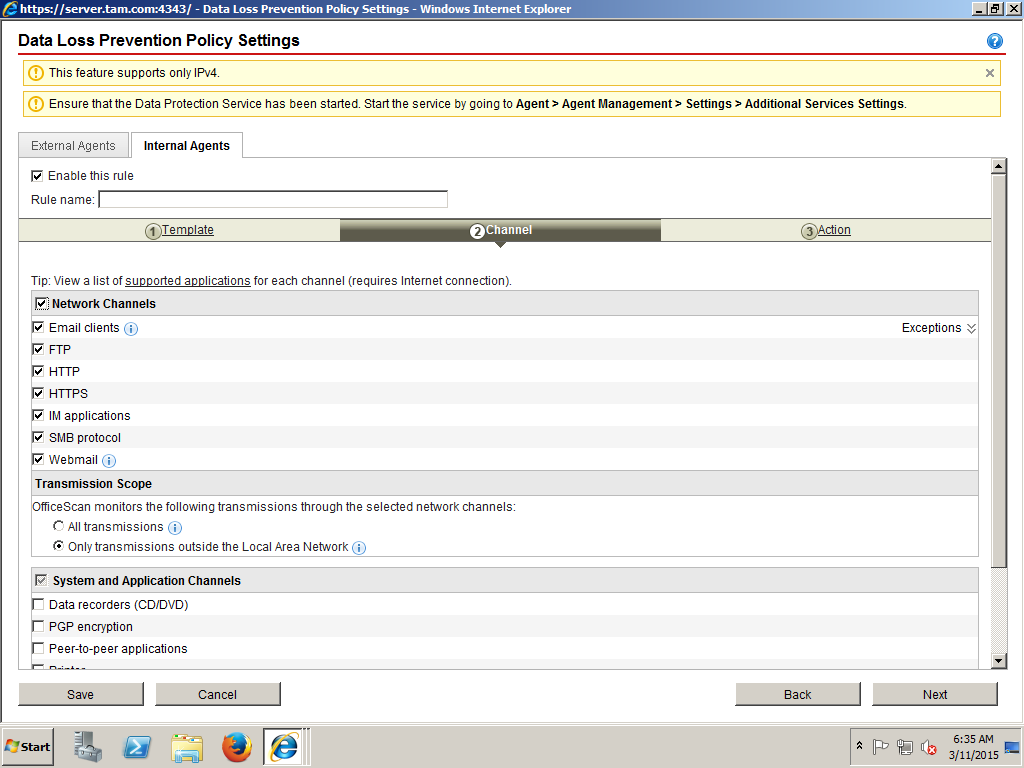
Với một plug-in “nhẹ”, bạn có thể đạt được tầm nhìn, kiểm soát các dữ liệu nhạy cảm và có thể ngăn chặn các thất thoát dữ liệu qua USB, email và web một cách dễ dàng, nhanh chóng. Plug-in DLP không yêu cầu các phần cứng hoặc hay phần mềm nào khác, và đó là đòn bẩy để xây dựng lĩnh vực và it leverages built-in regional and industry-specific templates to simplify deployment. Phần mềm tích hợp DLP cho phép chúng ta triển khai an toàn thông tin chỉ với một phần nhỏ chi phí và thời gian so với các giải pháp DLP truyền thống của các doanh nghiệp khác [9].

* Theo dõi mạng 24/7 với việc điều chỉnh theo thời gian thực
* Theo dõi và ghi lại các dữ liệu nhạy cảm truyền qua các điểm mạng
* Cho phép các IT hạn chế việc sử dụng các ổ đĩa USB, các thiết bị di dộng cổng USB, ghi đĩa CD/DVD, và các thiết bị di động khác.
* Theo dõi và xử lý với các dữ liệu sử dụng không đúng dựa trên từ khóa, regular expression và thuộc tính tập tin.
* Cải thiện cái nhìn và quyền kiểm sát với giải pháp tích hợp đầy đủ, quyền điều hành tập trung.
* Tốc độ kiểm tra và thực thi nắm bất dữ liệu và báo cáo thời gian thực.

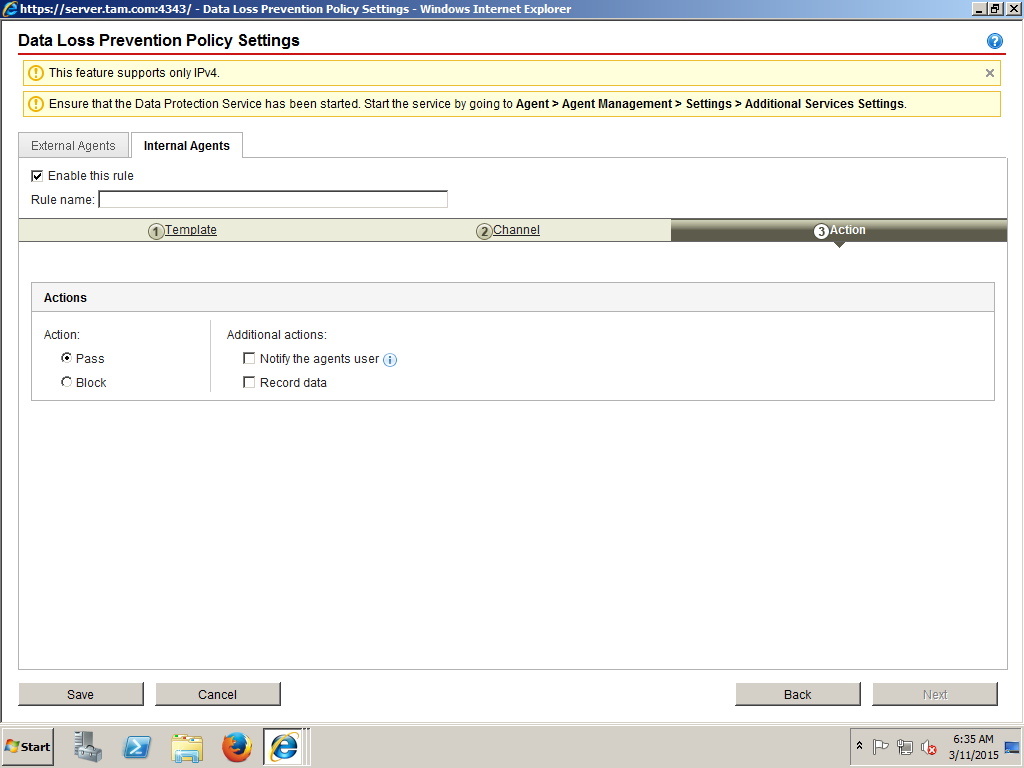
### Trial TrendMicro



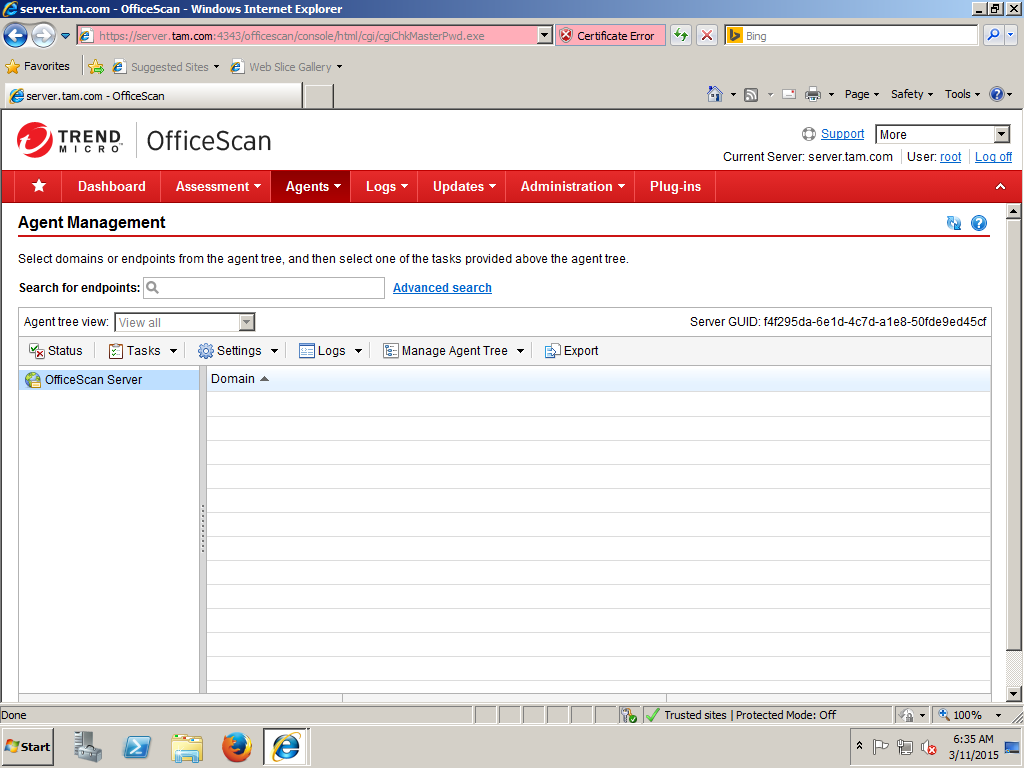
Hình ‑ Device Control Setting



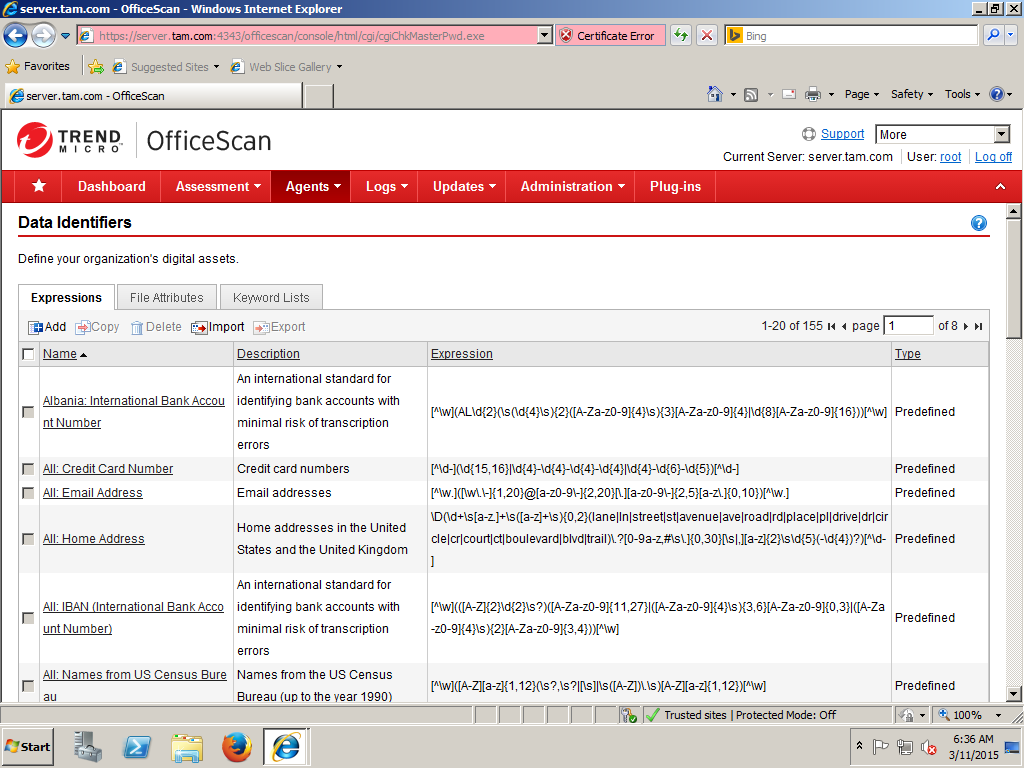
Hình ‑ Channel Configuration



Hình ‑ Action



Hình ‑ Agent Management



Hình ‑ Data Identifiers

## Symantec

### Giải pháp phần mềm

Symantec cung cấp rất nhiều phần mềm cho giải pháp bảo mật và lưu trữ. Những phần mềm này dễ dàng tích hợp với nhau [10].

* **Symantec Backup Exec System Recovery** – Data Loss Prevention tích hợp với Backup Exec System Recovery để quét các bản dự phòng (backup) dữ liệu quan trọng.
* **Symantec Control Compliance Suite** – Data Loss Prevention dữ liệu sự cố có thể được nạp vào Control Compliance Suite để xác định hệ thống có phải chịu trách nhiệm về chính sách kiểm soát kỹ thuật dựa trên các dữ liệu được lưu trữ trong đó.
* **Symantec Encryption** – Data Loss Prevention tích hợp với **Symantec Endpoint Encryption** để thực thi mã hóa dựa trên chính sách của các tập tin khi chúng được sao chép trong thời gian thực tới USBs; nó tích hợp với **Symantec FileShare Encryption** để giải nén, giải mã và phân tích văn bản trong các tài liệu mã hóa chia sẻ; nó tích hợp với **Symantec Universal Gateway Email** để thực thi mã hóa dựa trên chính sách của email và cung cấp xác nhận vòng kín bảo mật tại Data Loss Prevention Enforce Platform.
* **Symantec Endpoint Protection** – Data Loss Prevention tích hợp với Endpoint Protection để thực thi dựa trên chính sách lockdown (e.g., application control, port blocking, device control) trên máy tính và laptop. Với 1 năm hỗ trợ dành cho doanh nghiệp nhỏ, tương ứng 1 endpoint là 30 USD. Với 2 năm hỗ trợ, 1 endpoint tương ứng 50 USB. Con số này là 60 USD với 3 năm hỗ trợ.
* **Symantec Enterprise Vault** – Powered by the Data Loss Prevention công cụ phát hiện nội dung, **Enterprise Vault Data Classification Services** cung cấp dịch vụ dữ liệu phân loại phân loại dữ liệu và dịch vụ gắn thẻ chính sách định hướng cho email lưu trữ. Symantec Enterprise Vault Compliance Accelerator và Symantec Enterprise Vault Discovery Accelerator tận dụng các kết quả phân loại để hỗ trợ lưu trữ hồ sơ, tuân thủ pháp luật, và eDiscovery.
* **Symantec Messaging Gateway** – Data Loss Prevention tích hợp với Messaging Gateway vì vậy bạn có thể dễ dàng xem lại, phát hành và tuyến đường đi tin nhắn trực tiếp từ các Data Loss Prevention Enforce Platform.
* **Symantec Mobile Management** – Data Loss Prevention tích hợp với Mobile Management để thực thi các thiết lập mạng riêng ảo trên các thiết bị di động, màn hình hiển thị thông báo trên màn hình khi người dùng cố gắng để làm xáo trộn với hồ sơ thiết bị, và thực thi các điều khiển từ xa dựa trên chính sách trên các thiết bị.

### Các tính năng chống mất mát dữ liệu [11]

**Managerment and Reporting**

90% của DLP là về những thao tác sau khi bạn tìm dữ liệu quan trọng. Với Symantec Data Loss Prevention Enforce Platform, bạn có thể dễ dàng quản lý các chính sách và quy trình khắc phục hậu quả, xem xét sự cố, và giảm thiểu thiệt hại từ một nền tảng quản lý web thống nhất.

Enforce Platform là một giao diện điều khiển quản lý dựa trên web mạnh mẽ nơi bạn quản lý các chính sách và quy trình công việc mất mát dữ liệu, xem xét và khắc phục sự cố, phân tích và giảm thiểu rủi ro báo cáo và quản trị hệ thống. Nó bao gồm một module báo cáo nâng cao, Mất Symantec dữ liệu phòng chống IT Analytics, bạn có thể sử dụng dễ dàng tạo ra các báo cáo và biểu đồ mà giao tiếp ROI chương trình DLP của bạn tới giám đốc điều hành, các bên liên quan kinh doanh và kiểm toán viên.

**Laptops and Desktops**

Nhân viên được tải về và gửi dữ liệu bí mật trong khi ở văn phòng, trên đường đi hay ở nhà. Symantec Data Loss Prevention cho Endpoint monitors và bảo vệ dữ liệu được sử dụng trên máy tính xách tay và máy tính để bàn, khi người dùng đang ở trong và ngoài mạng công ty.

Quét Endpoint Khám phá và hàng tồn kho ổ đĩa cứng nội bộ trên máy tính xách tay và máy tính để bàn cho dữ liệu bí mật.

Endpoint Ngăn chặn theo dõi hoạt động của người dùng trên và ngoài mạng công ty; ngăn cản dữ liệu mật khỏi bị sao chép hoặc chia sẻ qua email không phù hợp, lưu trữ di động (ví dụ, USB, CD / DVD), in và fax, và lưu trữ đám mây (ví dụ, Dropbox). Các Endpoint Agent giám sát một loạt các sự kiện người sử dụng trên thiết bị đầu cuối vật lý đang chạy Windows XP, Windows Vista, Windows 7, và bây giờ Windows 8.1; phát hiện ra các dữ liệu được lưu trữ trên Mac OS X; giám sát máy tính để bàn ảo và các ứng dụng được lưu trữ bởi Citrix XenApp 6,5, VMware View và Microsoft Hyper-V; và màn hình dữ liệu chuyển qua Microsoft Remote Desktop Protocol (RDP).

**Mobile Devices**

Nhân viên được mang thiết bị riêng của họ để làm việc và truy cập dữ liệu bí mật từ họ, có hoặc không có sự cho phép của IT Security. Trong thực tế, một nghiên cứu mới tiết lộ rằng 2 trong số 5 nhân viên tải về tập tin làm việc với điện thoại thông minh và máy tính bảng cá nhân của họ. Hiện tại đội an ninh có thể quản lý Mang thiết bị riêng của bạn (BYOD) chính sách trong khi đảm bảo dữ liệu bí mật với Symantec dữ liệu phòng chống mất mát cho Mobile.

Mobile Email Monitor phát hiện email bí mật tải bởi người dùng iPad, iPhone và các thiết bị Android hiện nay thông qua giao thức Microsoft Exchange ActiveSync.

Màn hình điện thoại di động Ngăn chặn và bảo vệ mạng lưới truyền thông ra bên ngoài được gửi từ các khách hàng mail gốc, trình duyệt và các ứng dụng khác (ví dụ, Dropbox, Facebook) trên iPad và iPhone.

**Email và Web**

Email và web là hai trong số các kênh truyền hình phổ biến nhất cho sự mất mát dữ liệu. Một cuộc khảo sát gần đây cho thấy 50 phần trăm nhân viên thường xuyên gửi email tập tin làm việc từ văn phòng đến các tài khoản cá nhân. Symantec dữ liệu phòng chống mất mát cho mạng giám sát điểm ra và email đám mây để ngăn chặn dữ liệu bí mật bị phơi nhiễm qua giao thức mạng.

Ngăn chặn đám mây cho Microsoft Office 365 phát hiện dữ liệu bí mật trong email được gửi từ điện toán đám mây Microsoft Exchange Online. Nó liền mạch kéo dài chính sách mất dữ liệu hiện tại của bạn và công việc, do đó bạn có thể tự tin di chuyển các máy chủ Exchange của bạn để đám mây của Microsoft. Ngăn chặn đám mây cung cấp triển khai linh hoạt tại chỗ hoặc trong một cơ sở hạ tầng như một dịch vụ (IaaS) môi trường. Nó cũng hoàn toàn phù hợp với Symantec Email Security.cloud để đảm bảo cung cấp email thức.

Network Monitor dò tìm dữ liệu bí mật được gửi qua giao thức mạng có nguy cơ cao mà không cần lấy mẫu hoặc thả các gói tin: SMTP, HTTP, FTP, IM, NNTP, giao thức cổng cụ thể tùy chỉnh, và bây giờ Internet Protocol Version 6 (IPv6) mạng. Ngăn chặn cho mạng Email dò tìm dữ liệu bí mật trong email; thông báo cho người sử dụng vi phạm chính sách; và các khối hoặc các tuyến đường để gửi email mã hóa các cổng giao an toàn. Nó hỗ trợ tích hợp với bất kỳ chất độc SMTP-compliant Mail Transfer (MTA) và các dịch vụ đám mây như Microsoft Exchange Online và Symantec Email Security.cloud.

Ngăn chặn cho mạng Web dò tìm dữ liệu bí mật được gửi qua HTTP và HTTPS; thông báo cho người sử dụng vi phạm chính sách; và khối hoặc có điều kiện loại bỏ dữ liệu từ bài viết web. Nó hỗ trợ tích hợp với bất kỳ proxy Web ICAP tuân thủ và các dịch vụ đám mây như Google Postini và Symantec Web Security.coud.

**File Shares, Databases and Document Repositories**

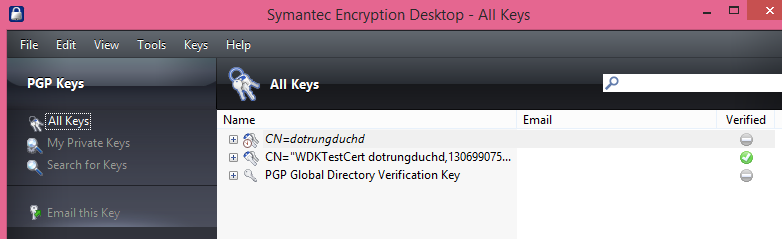
Người dùng được lưu trữ một lượng lớn các tập tin bí mật trên mạng công ty với quyền truy cập mở, mà làm cho họ dễ bị thiệt hại và trộm cắp. Symantec dữ liệu phòng chống mất cho lưu trữ quét trung tâm dữ liệu của bạn để khám phá và bảo vệ dữ liệu bí mật được lưu trữ trên tập tin chia sẻ, cơ sở dữ liệu, và các kho chứa. Dữ liệu Insight là một công nghệ giám sát tập tin duy nhất mà phân tích các mẫu truy cập và sử dụng tập tin trên mạng lưu trữ đính kèm (NAS) filers, các máy chủ Windows, và SharePoint. Nó xác định chủ sở hữu dữ liệu true; tính toán rủi ro folder để khắc phục được ưu tiên; tương quan chủ dữ liệu với sự cố lưu trữ; và cảnh báo bạn để hoạt động và người sử dụng outlier bất thường.

Khám phá mạng lưới thực hiện tốc độ cao chức năng quét của máy chủ tập tin, cổ phiếu, cơ sở dữ liệu, và kho tài liệu bao gồm cả Microsoft SharePoint và SharePoint Online, Documentum và LivelinkBREAK.

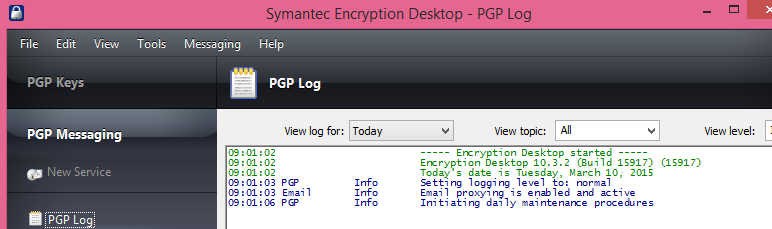
Mạng lưới Bảo vệ chặt tập tin bí mật bằng cách ly, di chuyển, hoặc áp dụng mã hóa và quản lý quyền kỹ thuật số dựa trên chính sách để các tập tin và thư mục tiếp xúc. Self-Service Portal Xử lý ô nhiễm cho phép chủ sở hữu dữ liệu kinh doanh để xem xét và khắc phục các vi phạm chính sách tập tin mạng trực tiếp từ một cổng thông tin trực tuyến trực quan, và sắp xếp hợp lý các quy trình xử lý rủi ro.

### Trial Symantec File Share Encryption

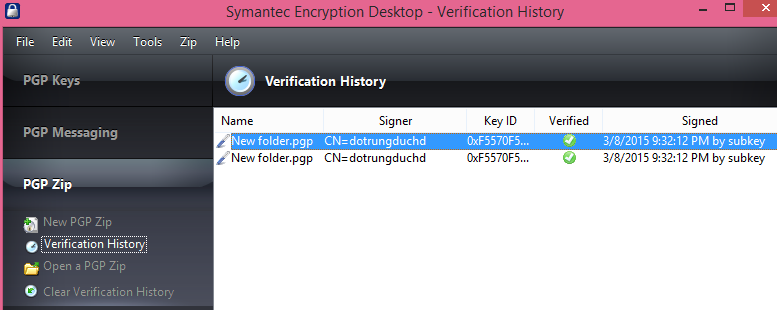
Giao diện Symantec Encryption Desktop



Hình ‑ Quản lý key

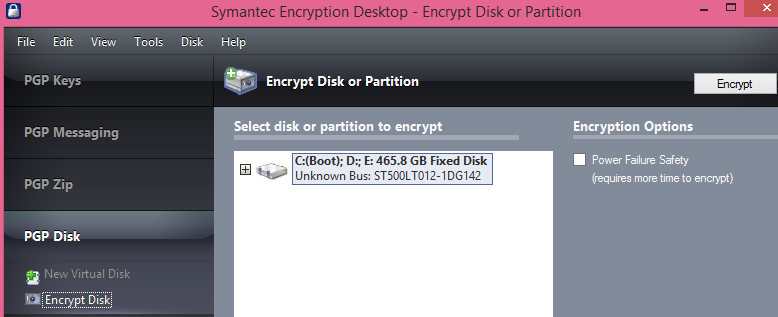


Hình ‑ Lưu lịch sử hoạt động



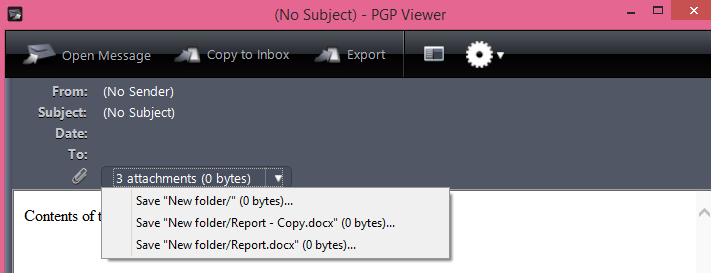
Hình ‑ Mã hóa file dạng Zip

Ví dụ: thư mục new folder được mã hóa bởi user dotrungduchd



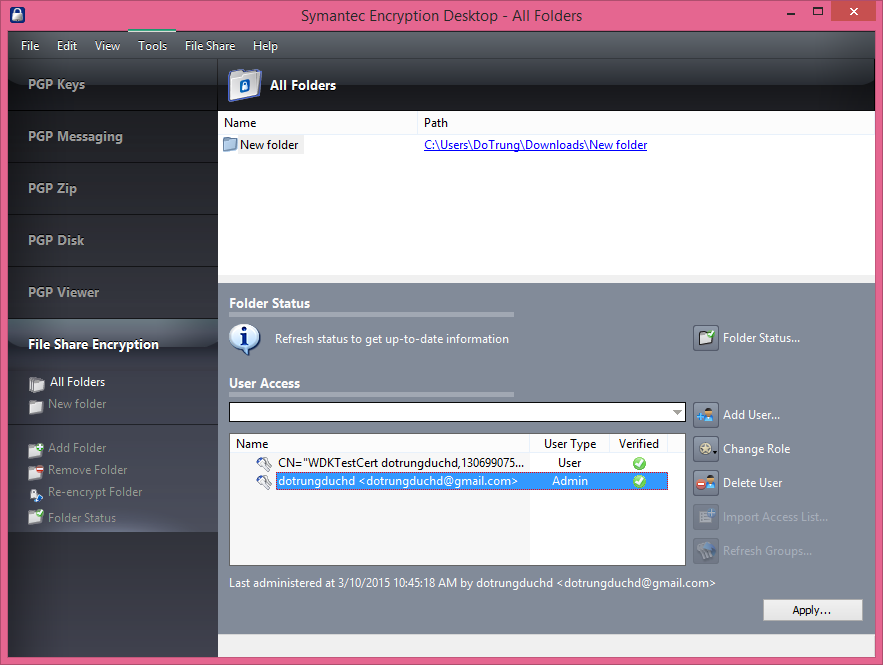
Hình ‑ Mã hó ổ đĩa

Ví dụ: Ổ đĩa C



Hình ‑ Xem thông tin file mã hóa

Ví dụ: Xem thông tin file “new folder” đã được mã hóa



Hình ‑ Chia sẻ file mã hóa

Ví dụ: folder: new folder đã được mã hóa được chia sẻ cho 2 user, trong đó user dotrungduchd là admin

## RSA

### Giải pháp phần mềm

Khám phá và theo dõi vị trí và luồng của dữ liệu nhạy cảm như dữ liệu thẻ tín dụng của khách hàng, nhân viên PII, hoặc tài sản trí tuệ của doanh nghiệp. Hướng dẫn người dùng cuối và thực thi kiểm soát để ngăn ngừa mất dữ liệu nhạy cảm thông qua email, web, máy tính, điện thoại thông minh, và nhiều hơn nữa [12].

### Các tính năng chống mất mát dữ liệu [12]

**Tổng quan**

Ngăn chặn việc mất dữ liệu nhạy cảm thông qua nhiều nguy cơ rủi ro, bao gồm email, webmail, phương tiện truyền thông xã hội, FTP, Web, Web 2.0, máy tính, máy ảo, điện thoại thông minh, Microsoft SharePoint, máy chủ tập tin, NAS / SAN, cơ sở dữ liệu, và các thiết bị USB.

**Phân loại chính xác**

Xác định các dữ liệu nhạy cảm với độ chính xác công nghiệp tốt nhất thông qua sự kết hợp RSA DLP của nhận thức khoa học dựa trên phân loại nội dung, vân tay dựa trên máy móc, phân tích siêu dữ liệu phong phú, và các chính sách chuyên gia mục đích xây dựng.

**Giáo dục người dùng**

Dựa vào RSA DLP để theo dõi các hành động người dùng lấy dữ liệu nhạy cảm và giáo dục họ trong thời gian thực về các vi phạm chính sách. Điều này cải thiện nhận thức về nguy cơ trong những người dùng cuối, ảnh hưởng đến hành vi của họ trong việc đối phó với các dữ liệu nhạy cảm.

**Nền tảng tích hợp**

Tích hợp RSA DLP với nhiều nền tảng doanh nghiệp, bao gồm cả Microsoft, Cisco, EMC, VMware, Citrix, McAfee, Symantec, Trend Micro, và Blue Coat, để tối đa hóa việc sử dụng các cơ sở hạ tầng hiện tại của bạn.

**Tự động hóa công việc**

Cải thiện quản lý dự án DLP bằng cách tận dụng một quy trình làm việc tự động người- và quá trình làm trung tâm cho chính sách quản lý, khắc phục sự cố, quản lý và báo cáo.

## WebSense

### Giải pháp phần mềm [13]

**TRITON AP-**WEB: Có được sự bảo về thời gian thực chống lại các Advanced Threats và việc đánh cắp dữ liệu với nhiều phương thức triển khai cho cả on-premise và người dùng từ xa.

* Chặn các Advanced Threats – TRITON AP-WEB nhận dạng các Zero-day malware và các hoạt động gây hại khác trên toàn bộ Kill chain để ngăn chặn các mối đe dọa mới nhất.
* Bảo vệ toàn diện – Thống nhất hệ thống bảo vệ inbound va outbound Web, chặn các mối đe dọa và cung cấp actionable insight.
* Ngăn chặn đánh cắp dữ liệu – Một engine được tích hợp đầy đủ theo chuẩn DLP doanh nghiệp đơn giản hóa việc an ninh thông tin và compliance bao gồm OCR, “Drip DLP” và các tính năng độc đáo khác.
* Cho phép truy cập liên tục tới tài nguyên Web, điều khiển và làm việc với thiết bị di động, biết rằng đó là dữ liệu nhạy cảm còn lại.

TRITON AP-EMAIL: Xác định các mục tiêu bị tấn công, những người dùng có nguy cơ cao và các mối đe dọa bên trong đồng thời cho phép các thiết bị di động làm việc và an toàn như công nghẹ của Microsoft Office 365™.

* Ngăn chặn APT và các mối đe dọa các khác.
* Bảo mật các dữ liệu nhạy cảm khỏi các cuộc tấn công bên ngoài và mối đe dọa bên trong.
* Áp dụng các công nghệ mới như Microsoft Office 365 và hỗ trợ cho các nhân viên từ xa một cách an toàn.
* Giúp do người dùng nâng cao ý thức và xác định các hành động có nguy cơ gây hại.

TRITON AP-DATA: Phát hiện và bảo vệ những dữ liệu nhảy cảm bất cứ nơi nào chứa nó – trên servers, endpoints hoặc các dich vụ đám mây như MO365, cũng như trong sử dụng hay chuyển đổi qua Web hay email gateways.

* Áp dụng dịch vụ đám mây như MO365 và Box mà không sợ bị đánh cắp dữ liệu
* Deploy easy-to-use security controls for compliance and regulatory requirements for auditors as well as for executives and investors.
* Phát hiện các dữ liệu nhạy cảm bên trong hình cũng như trong các dữ liệu được scan và ảnh chụp màn hình dễ dàng.
* Thống nhất các giải pháp bảo mật, các chính sách bảo vệ, chia sẻ thông tin theo nhiều điểm và tập trung quyền kiểm soát cho an toàn thông tin.

TRITON AP-ENDPOINT: Protect roaming người dùng khỏi việc đánh cắp dữ liệu và nắm quyền kiểm soát thông tin nhạy cảm trên các hệ thống Mac OS X và Windows trong và ngoài mạng.

* Phát hiện và bảo vệ các dữ liệu quan trọng khỏi đánh cắp khi người dùng ở trong và ngoài mạng công ty.
* Bảo mật các endpoint Mac OS X và Windows, bao gồm chống thoát dữ liệu qua USB và các thiết bị lưu trữ khác.
* Safely embrace cloud services like social media, Microsoft Office 365™ and Box through data visibility and control.
* Nhanh chóng đáp ứng các yêu cầu cần tuân thủ, quy định với một thư viện rộng lớn đầy các chính sách và thỏa mãn các auditors với các mẫu báo cáo chuẩn hay tùy chỉnh.
* Kiến trúc TRITON cung cấp cách quản lý thống nhất của giải pháp TRITON, củng cố các tasks để các nhân viên IT có thể tập trung cho các dự án khác.

Giá thành của các sản phẩm sẽ được thỏa thuận giữa **WebSense Partner** và người mua.

## Kết luận

Các giải pháp bảo vệ dự liệu trên thiết bị lưu trữ của các công ty bảo mật thường áp dụng theo mô hình server – endpoint (client). Những chính sách bảo mật được server đưa ra và bắt buộc endpoint phải tuân theo. Nhưng chính điều đó gây ra sự mất tự nhiên, cảm giác làm chủ thiết bị của mình không còn. Thay vào đó là một loạt các chính sách cưỡng bức người dùng như chặn USB, chỉ cho phép đọc USB,…

Hơn thế nữa, những giải pháp này áp dụng cho những công ty quy mô lớn, giá thành rất cao. Trong khi cá nhân và các mô hình vừa và nhỏ chưa có giải pháp thay thế nào phù hợp ngoài việc thao tác thủ công mã hóa dữ liệu trước khi lưu trữ ra các thiết bị nhớ ngoài.

# Can thiệp hệ thống

* + Nội dung Chương 3 trình bày các biện pháp can thiệp vào hệ thống để chặn và lấy thông tin cần thiết phục vụ cho giải pháp bảo mật sau này. Chương này trình bày 2 nội dụng chính:

- Hook API

- Microsoft Office Add-in

## Hook API

* Vấn đề:

Trong quá trình xử lý, hệ thống gọi các hàm API để thực hiện các chức năng nhất định. Các hàm API khi thực hiện sẽ xuất hiện sự kiện và thông báo tương ứng. Vấn đề chính của chúng ta là làm cách nào để biết được sự kiện khi nào một hàm API được gọi và thực thi? Làm sao để thay đổi hoặc thêm một vài xử lý trước khi hàm kết thúc?

* Giải pháp

Sử dụng Hook API để giải quyết vấn đề. Hook API là kỹ thuật được sử dụng để theo dõi ứng dụng và các hàm API của hệ thống.

Hook API giúp lập trình viên theo dõi và kiểm soát các hành động xảy ra bởi các hàm API. Sử dụng để xác định và sửa lỗi, thay đổi và phát triển chức năng của hàm API.

Trong giải pháp này có 2 phần quan trọng đó là chèn DLL vào process (inject) và chặn hàm API (intercept).

## Kỹ thuật chèn DLL vào process

### Registry

* Vấn đề

Làm cách nào để chèn DLL vào nhiều chương trình khi chúng tải các thư viện liên kết động (DLL).

* Giải pháp

Có thể chèn DLL vào process liên kiết với USER32.DLL bằng cách thêm tên của DLL đó vào trong registry : ***HKEY\_LOCAL\_MACHINE\Software\Microsoft\***

***Windows NT\CurrentVersion\Windows\AppInit\_DLLs.***Khi đó những ứng dụng cơ bản của Windows sẽ tải DLL được quy định trong registry lên giống như một phần của khởi tạo USER32.DLL. Trong hàm DllMain của USER32 thực hiện lệnh gọi hàm LoadLibrary() cho những giá trị trong thanh ghi trên. Hạn chế của kỹ thuật này là chỉ hỗ trợ cho Windows NT và Windows 2k. Một số lưu ý:

* Khởi động lại Windows để kích hoạt hoặc dừng chèn DLL vào process.
* Chỉ có tác dụng với những process sử dụng USER32.DLL, vì thế có thể một số ứng dụng không thực hiện được.
* Không kiểm soát được khi chèn DLL vào process. Nó có tác dụng với tất cả ứng dụng GUI nên có thế có nhiều chi phí phát sinh.

### System-wide Windows Hooks

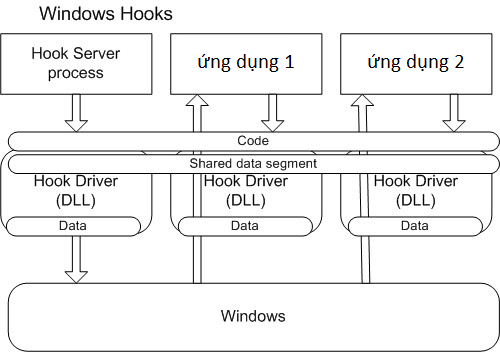
* Vấn đề

Khi chúng ta có nhu cầu xử lý Hook trên toàn bộ hệ thống. Vấn đề đặt ra là làm sao có thể chèn DLL vào tất cả process đang chạy (ngoại trừ các service của windows).

* Giải pháp

Sử dụng System-wide Windows Hooks.

Đây là một kỹ thuật rất phổ biến để chèn DLL vào process. Ứng dụng có thể cài một hàm lọc để theo dõi message trong hệ thống và xử lý chúng trước khi tới windows procedure. Sử dụng hàm **SetWindowsHookEx**() để cài đặt hook cùng các tham số tương ứng. Khi cài đặt hook thành công, hệ điều hành sẽ ánh xạ DLL vào địa chỉ của mỗi process. Những biến trong DLL sẽ được tự xử lý và không thể chia sẻ cho những process khác. Mọi biến chứa dữ liệu cần chia sẻ phải ở trong ‘*share data section*’.



Hình ‑ Hook Server can thiệp vào 2 ứng dụng [14]

Sau khi hàm SetWindowsHookEx() thực hiện thành công, hệ điều hành tự động chèn DLL vào tất cả các process đáp ứng yêu cầu của hook filter. Một khi đã chèn DLL vào trong process, sử dụng hàm UnhookWindowsHookEx() để unload DLL hoặc tắt ứng dụng hook. Khi gọi hàm UnhookWindowsHookEx(), hệ điều hành lặp qua danh sách những process bị chèn DLL và giảm DLL’s lock count đi. DLL chèn vào sẽ tự xóa khỏi không gian vùng nhớ của process.

Ưu điểm:

* Hỗ trợ Windows NT/2K và họ Windows 9x hy vọng sẽ hỗ trợ các phiên bản Windows tiếp theo.
* Chủ động gọi hàm UnhookWindowsHookEx() khi không cần thiết phải Hook nữa.

Nhược điểm:

* Có thể làm giảm đáng kể hiệu suất của hệ thống vì làm tăng xử lý cho mỗi message.
* Ảnh hưởng đến việc xử lý của toàn hệ thống và có thể phải khởi động lại để khôi phục lỗi.

### Injecting DLL by using CreateRemoteThread() API function

* Vấn đề

Cài đặt và hook trên nhiều tiến trình không cần thiết có thể làm chậm hệ thống và gây ra những lỗi nghiêm trọng. Liệu có cách nào chỉ xử lý trên 1 số tiến trình xác định? Khi đó, chúng ta cần một giải pháp mới tập trung vào xử lý một tiến trình.

* Giải pháp

Sử dụng CreateRemoteThread() để tạo tiểu trình trong tiến trình xác định và tải DLL vào tiến trình mục tiêu.

Về cơ bản, mỗi tiến trình thực hiện tải DLL động bằng hàm **LoadLibrary**(). Vấn đề là làm sao để gọi hàm load DLL từ bên ngoài tiến trình? Ý tưởng là tạo một thread con trong tiến trình đó và load DLL từ bên ngoài vào trong tiến trình. Để hiện thực hóa ý tưởng này, chúng ta có hàm **CreateRemoteThread**() và **CreateRemoteThreadEx**() sẽ tạo một thread chạy trong vùng nhớ ảo của tiến trình.

HANDLE WINAPI CreateRemoteThread(

\_In\_   HANDLE hProcess,

\_In\_   LPSECURITY\_ATTRIBUTES lpThreadAttributes,

\_In\_   SIZE\_T dwStackSize,

\_In\_   LPTHREAD\_START\_ROUTINE lpStartAddress,

\_In\_   LPVOID lpParameter,

\_In\_   DWORD dwCreationFlags,

\_Out\_  LPDWORD lpThreadId

);

Các tham số cần chú ý:

*lpStartAddress* [in]: Một con trỏ tới hàm ứng dụng định nghĩa kiểu **LPTHREAD\_START\_ROUTINE**  sẽ được thực thi bởi thread mới tạo và đóng vai trò là địa chỉ bắt đầu của thread từ xa.

A pointer to the application-defined function of type **LPTHREAD\_START\_ROUTINE** to be executed by the thread and represents the starting address of the thread in the remote process. The function must exist in the remote process. For more information, see [**ThreadProc**](https://msdn.microsoft.com/en-us/library/windows/desktop/ms686736(v=vs.85).aspx).

*lpParameter* [in]: Con trỏ tới một biến truyền tới hàm sẽ được thực thi bởi thread mới.

A pointer to a variable to be passed to the thread function.

Ví dụ: tiến hành chèn Hook.dll vào ứng dụng có handle là hProcessForHooking với tham số "C:\\Hook.dll"

hThread = ::CreateRemoteThread(

hProcessForHooking,

NULL,

0,

pfnLoadLibrary,

"C:\\Hook.dll",

0,

NULL);

* Trong đó pfnLoadLibrary là con trỏ hàm của LoadLibrary. Sử dụng hàm GetProcAddress() để lấy địa chỉ này. May mắn là trong Kernel32.DLL luôn ánh xạ địa chỉ giống nhau ở tất cả process nên địa chỉ hàm lấy được luôn hợp lệ.
* Tham số ngay sau đó là đường dẫn tới DLL cần load: "C:\\Hook.dll ".

Một lưu ý quan trọng khi sử dụng phương pháp này đó là truyền cờ PROCESS\_ALL\_ACCESS khi OpenProcess(). Điều này giúp ta có đầy đủ quyền truy cập trong process.

There is an important thing we should consider, if implanting through CreateRemoteThread() API. Every time before the injector application operate on the virtual memory of the targeted process and makes a call to CreateRemoteThread(), it first opens the process using OpenProcess() API and passesPROCESS\_ALL\_ACCESS flag as parameter. This flag is used when we want to get maximum access rights to this process. In this scenario OpenProcess() will return NULL for some of the processes with low ID number. This error (although we use a valid process ID) is caused by not running under security context that has enough permissions. If you think for a moment about it, you will realize that it makes perfect sense. All those restricted processes are part of the operating system and a normal application shouldn't be allowed to operate on them. What would happen if some application has a bug and accidentally attempts to terminate an operating system's process? To prevent the operating system from that kind of eventual crashes, it is required that a given application must have sufficient privileges to execute APIs that might alter operating system behavior. To get access to the system resources (e.g. smss.exe, winlogon.exe, services.exe, etc) through OpenProcess() invocation, you must be granted the debug privilege. This ability is extremely powerful and offers a way to access the system resources, that are normally restricted. Adjusting the process privileges is a trivial task and can be described with the following logical operations:

* Open the process token with permissions needed to adjust privileges
* Given a privilege's name "SeDebugPrivilege", we should locate its local LUID mapping. The privileges are specified by name and can be found in Platform SDK file winnt.h
* Adjust the token in order to enable the "SeDebugPrivilege" privilege by callingAdjustTokenPrivileges() API
* Close obtained by OpenProcessToken() process token handle

For more details about changing privileges see [10] "Using privilege".

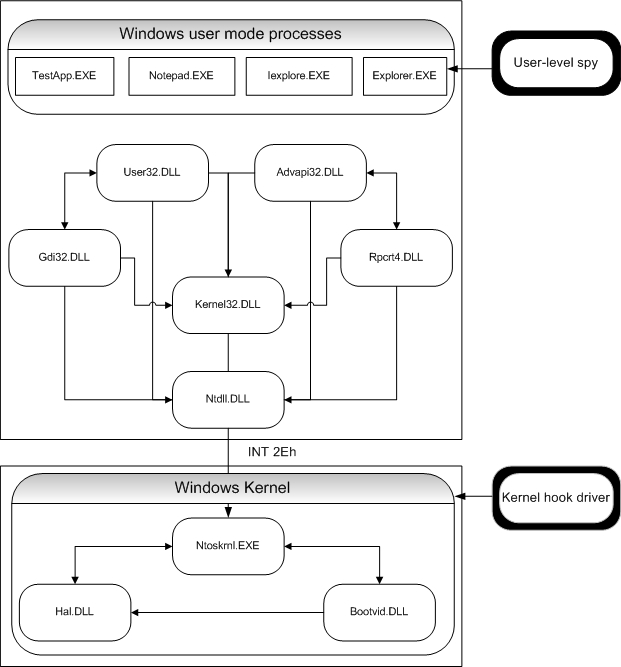
## Cơ chế chặn hàm API

* Vấn đề

Giả sử chúng ta đã chèn được DLL vào process cần xử lý. Vấn đề tiếp theo là làm thế nào để biết được hàm API xảy ra khi nào và tiến hành thay đổi xử lý bên trong hàm API.

* Giải pháp

Trước hết, chúng ta cần hiểu là Windows có 2 chế độ là User mode và Kernel mode. User mode là chế độ mà hệ điều hành cho phép chúng ta dễ dàng can thiệp và xử lý. Các thao tác của người dùng sẽ được các phương thức ở mức User mode xử lý và gọi xuống các hàm ở Kernel. Kernel mode đòi hỏi lập trình viên phải viết và xử lý trên Driver. Đối với hệ điều hành 64 bit thì Driver phải được Microsoft ký mới được thực thi các chức năng trong đó.



Hình ‑ User mode và Kernel mode trong Windows [14]

Chúng em chủ yếu tìm hiểu về Win32 User Level hooking.

### Windows subclassing

Sử dụng phương thức **SetWindowLongPtr**() với tham số **GWLP\_WNDPROC** để truyền con trỏ tới windows procedure mới. Khi đã cài đặt một subclass, mỗi khi có message tới windows được chỉ định (A) nó sẽ tìm đến địa chỉ của windows procedure liên kết với windows tương ứng (A) và gọi procedure của bạn thay vì hàm gốc.

Giới hạn: Chỉ hợp lệ với các process đặc biệt và ứng dụng không nên có subclass của windows tạo bởi process khác.

Thường sử dụng hook thông qua add-in và có thể lấy được handle của window mà bạn muốn thay thế procedure của nó.

### Proxy DLL

Thay thế một DLL cùng tên và export tất cả các tính năng của DLL gốc. Kỹ thuật này dễ dàng cài đặt bằng function forwarders. Sử dụng #pragma comment :

#pragma comment(linker, "/export:DoSomething=DllImpl.ActuallyDoSomething")

### Code overwriting

Đây là phương pháp hook rất phổ biến được áp dụng như là một trong những phương pháp đầu tiên để chặn hàm API.

Bước đầu tiên, chúng ta phải xác định chính xác địa chỉ hàm API trong bộ nhớ. Sử dụng hàm GetProcessAddress() để lấy địa chỉ hàm cần hook.

Lưu địa chỉ hàm gốc lại để có thể khôi phục lại hoặc sử dụng trong quá trình hook.

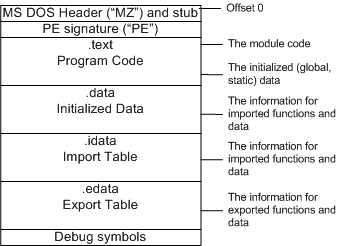
Ghi đè một lệnh JMP vào địa chỉ hàm gốc để chuyển hướng đến hàm tự định nghĩa (hook function).

Trong hàm tự định nghĩa, chúng ta toàn quyền xử lý dữ liệu tham số đã hook được.

Gọi lại hàm gốc ban đầu nếu cần thiết và khôi phục lại địa chỉ hàm gốc khi hoàn thành hook.

### Spying by altering of the Import Address Table

Đầu tiên, chúng ta phải biết về Portable Excutable (PE) Windows file format.



Hình ‑ PE format khi thực thi chương trình [14]

PE format là cấu trúc dữ liệu đóng gói thông tin cần thiết cho Windows OS loader quản lý mã thực thi [15]. Nhưng thông tin này bao gồm dữ liệu DLL, API export và import table, thông tin quản lý tài nguyên và thread-local-storage (TLS). Trên hệ điều hành NT, PE format sử dụng cho EXE, DLL, SYS (device driver) và các kiểu file khác [15].

Trong đó, Import Address Table (IAT nằm trong phần .idata) sẽ lưu địa chỉ hàm của module binary (DLL) khác. Mỗi khi ứng dụng cần gọi hàm sẽ tìm trong bảng này. Như thế sẽ tiết kiệm hơn khi ứng dụng cần sử dụng các hàm import.

Thay đổi địa chỉ hàm trong IAT sẽ giúp chuyển hướng tới hàm hook của mình.

## Microsoft Office Add-in

Mã hóa Word Add-in.

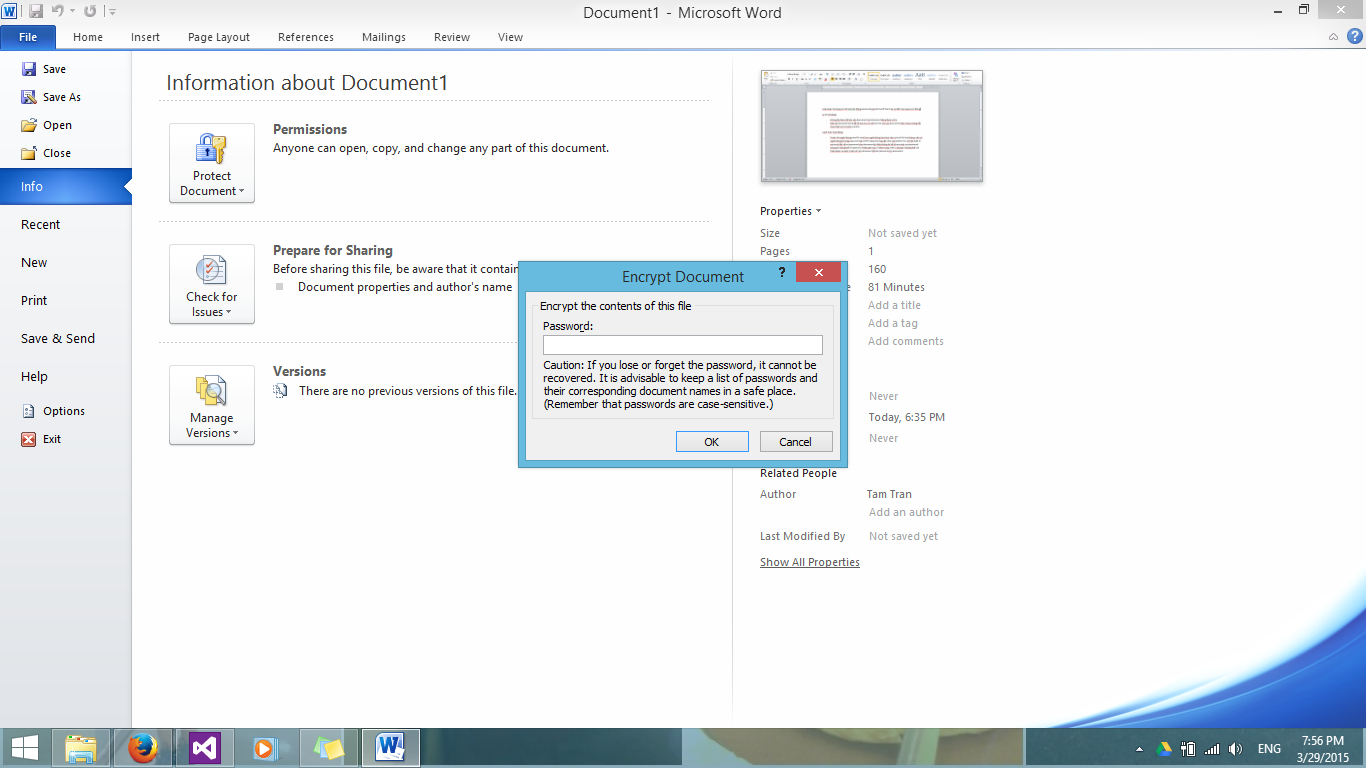
### Sử dụng cơ chế mã hóa mặc định của Word

* Vấn đề

Dữ liệu văn phòng thường được soạn thảo bằng Microsoft Word. Vấn đề đặt ra : cần mã hóa dữ liệu Word trước khi thoát chương trình.

* Giải pháp:

Sử dụng cơ chế lock file bằng password của Microsoft Word và thể hiện nó một cách trực quan sinh động hơn cho người dùng.



Hình ‑ Mã hóa dữ liệu của Microsoft Word

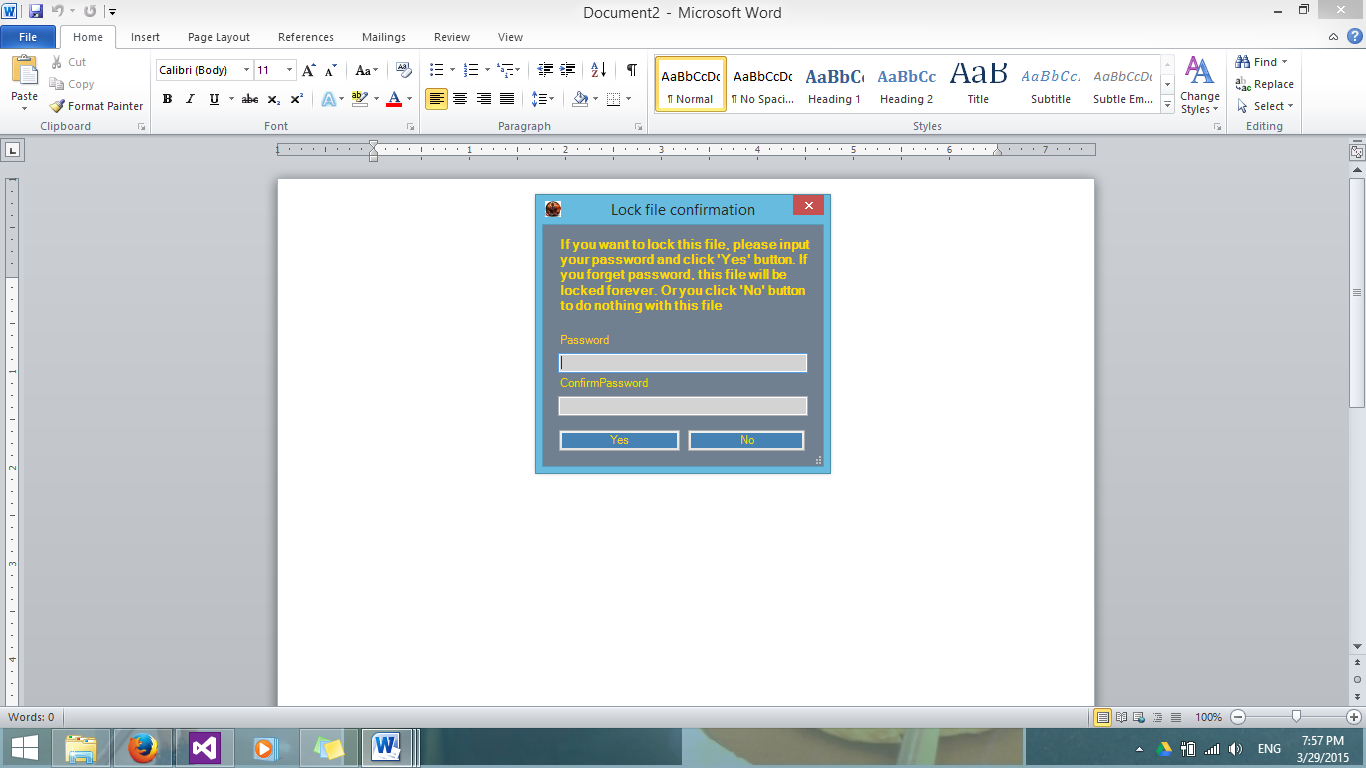
**Lý do sử dụng:**

Không lấy được dữ liệu của document khi document đang được xử lý.

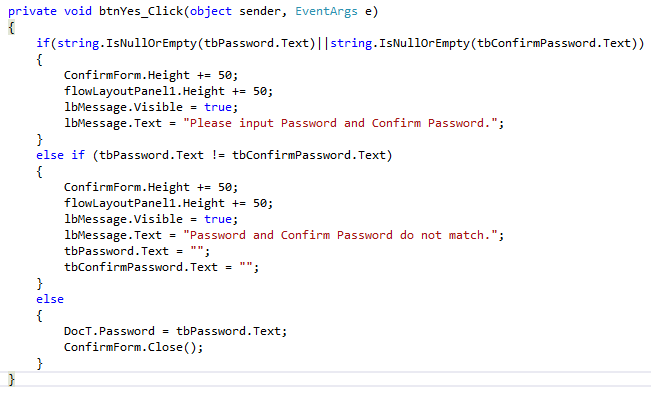
Mặc dù Microsoft Word đã hỗ trợ cho cơ chế lock file cho các document khác nhau nhưng vẫn chưa thật sự tự nhiên (smooth).

**Cách thức hoạt động:**

Trước khi người dùng save file ta sẽ cho người dùng lựa chọn việc có lock file hay không, nếu có người dùng sẽ nhập **password** và **confirm password** sau đó click vào button Yes. Khi đó Add-In mà ta cài đặt sẽ set password cho document ấy. Nếu không thì sẽ bỏ qua việc set password.



Hình ‑ Giao diện mới khi lock file

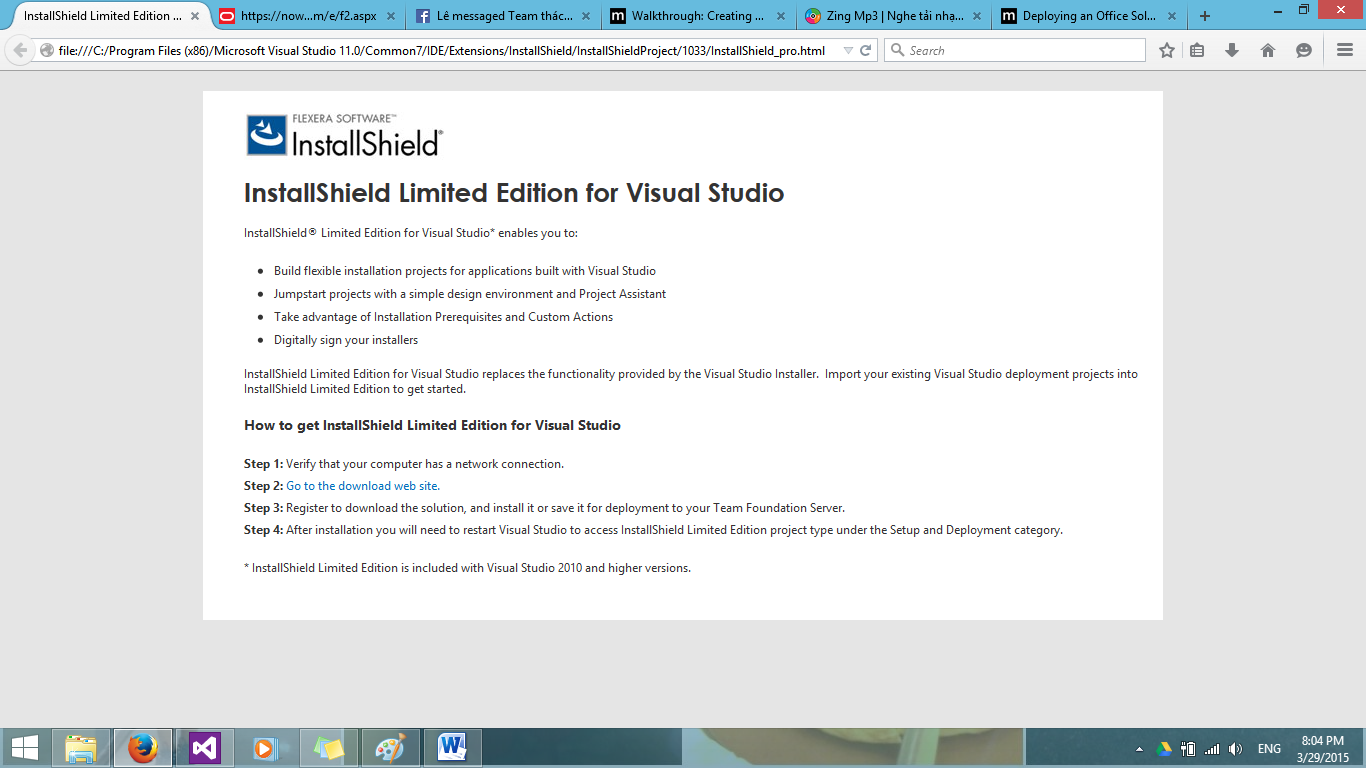


Chúng ta không thể set password một cách tùy ý ở bên trong code vì chúng ta không thể can thiệp được sự kiện trước khi mở document để set password cho document.

**Cài đặt:**

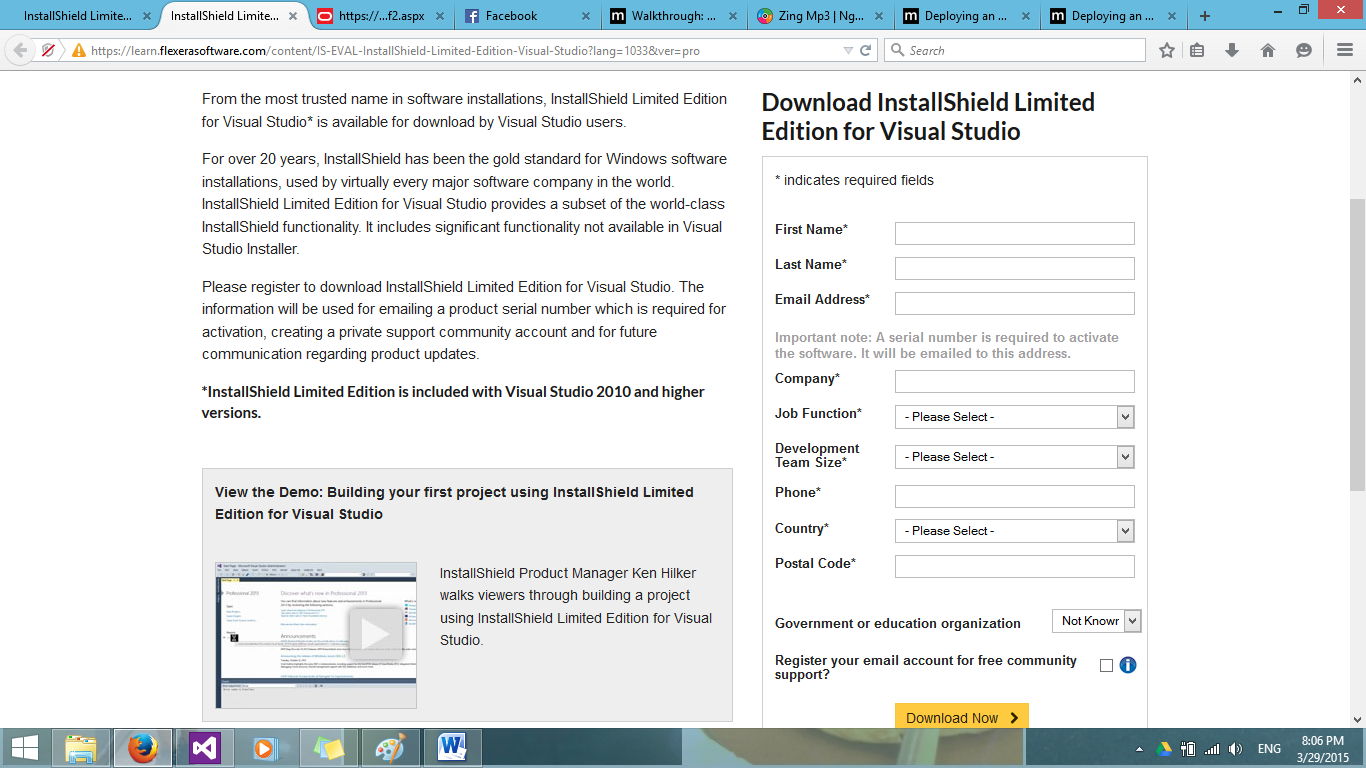
Hiện tại sẽ sử dụng Windows Installer. Làm theo các bước được hướng dẫn theo link sau: <https://msdn.microsoft.com/en-us/library/cc442767%28v=vs.110%29.aspx>

Nhưng tồn tại vấn đề sau: Khi tạo project **Enable InstallShield Limited Edition** thì sẽ hiện ra một link sau



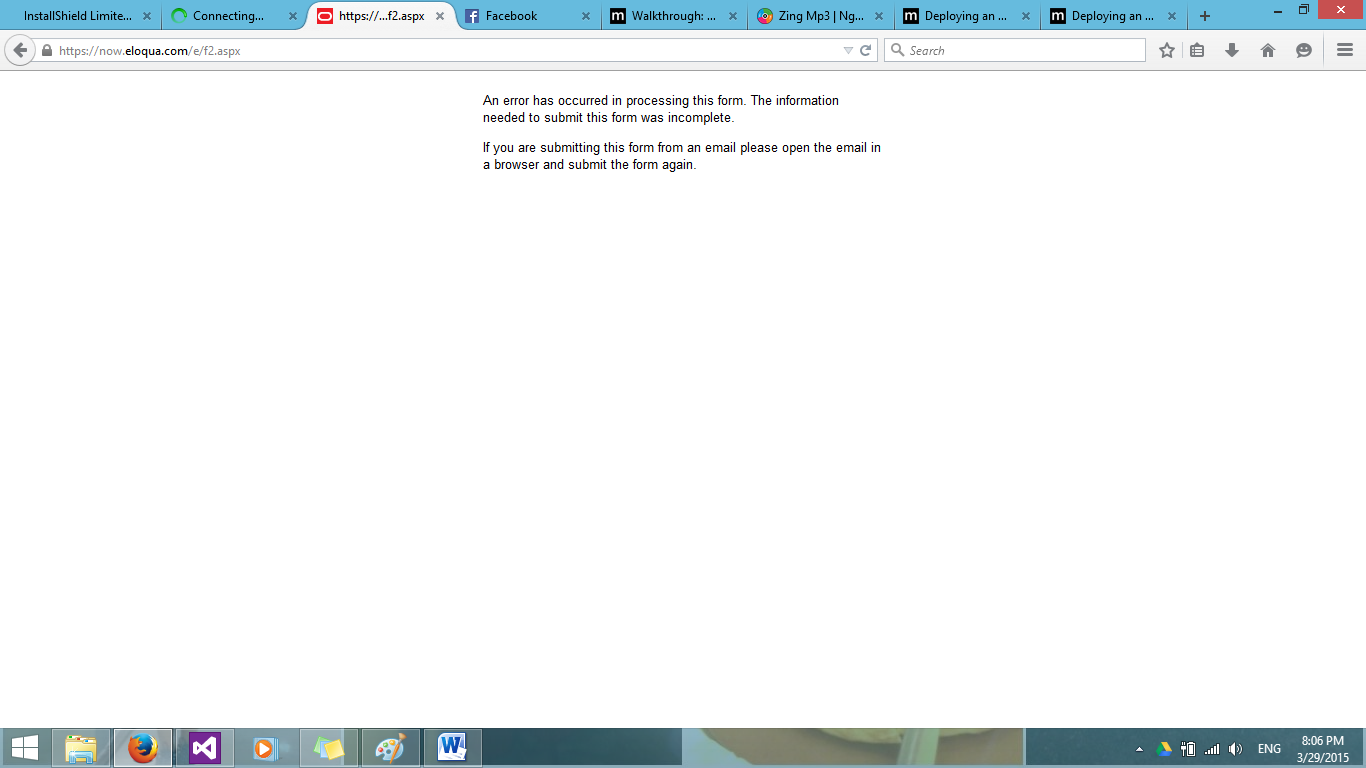
Hình ‑ InstallShield Limited Edition

Click theo đường dẫn để đăng ký để được download nhưng bước đăng ký vẫn không hoàn thành được mặc dù đã thử nhiều lần.



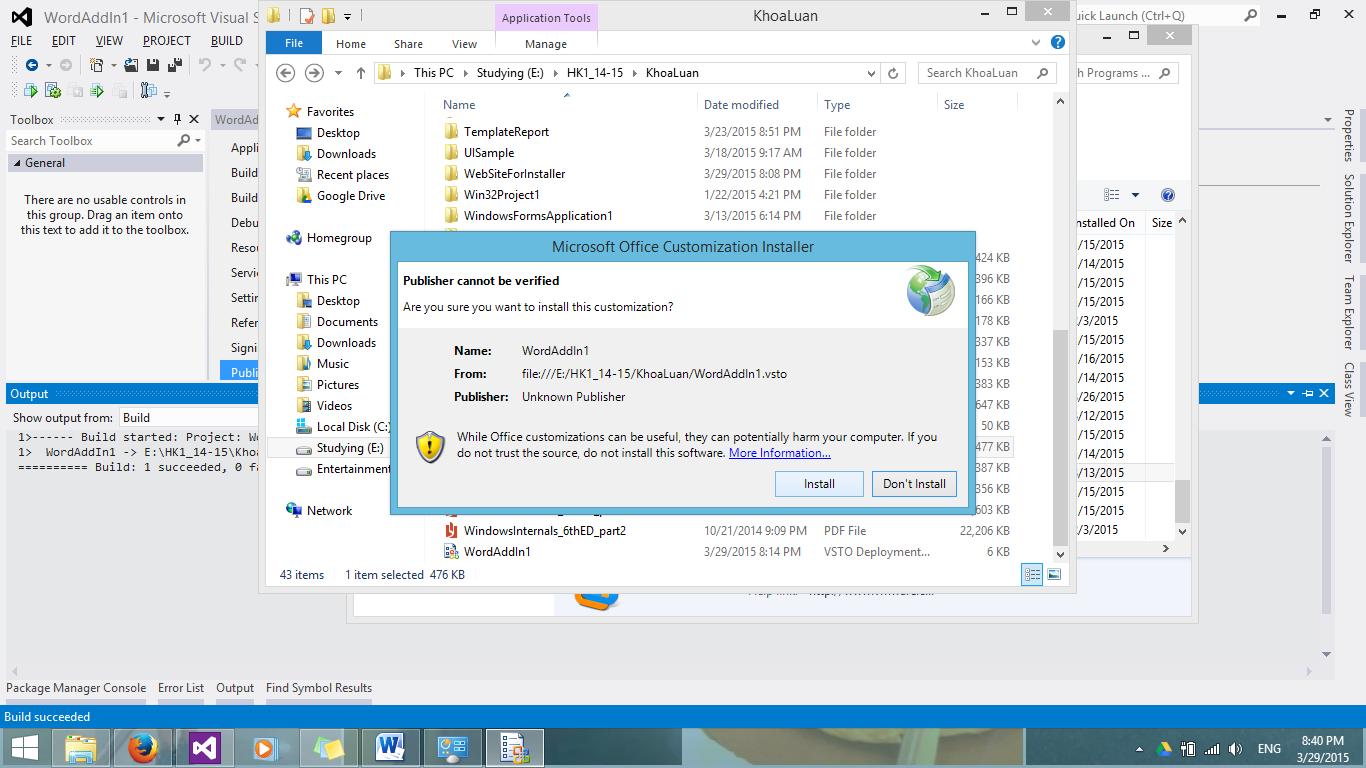
Hình ‑ InstallShield Limited Edition Download

Lỗi.

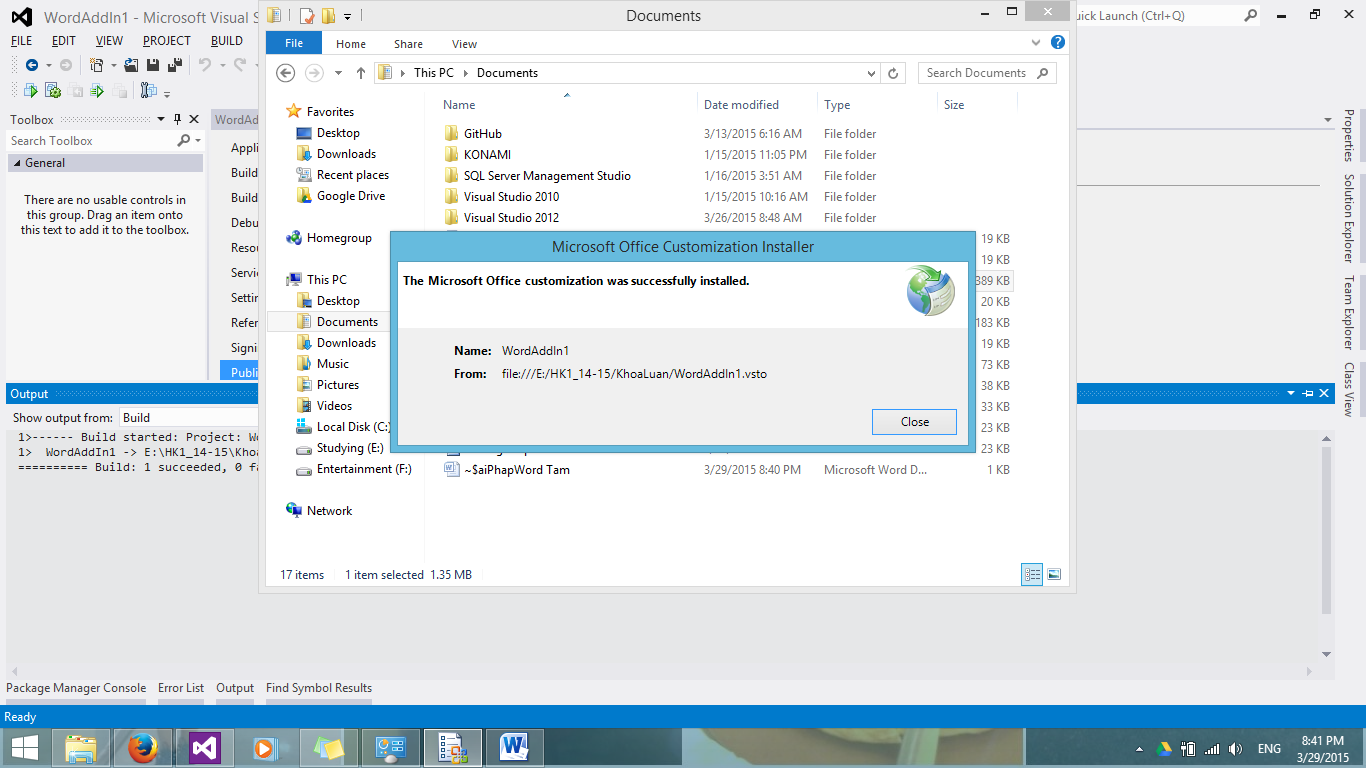


Hình ‑ InstallShield Limited Edition Error

Về cách sử dụng ClickOne <https://msdn.microsoft.com/en-us/library/bb772100%28v=vs.110%29.aspx> theo đường dẫn vừa nêu thì thành công và được kết quả sau nhưng hiện tại chỉ chạy được local: ?



Hình ‑ Cài đặt WordAddin1

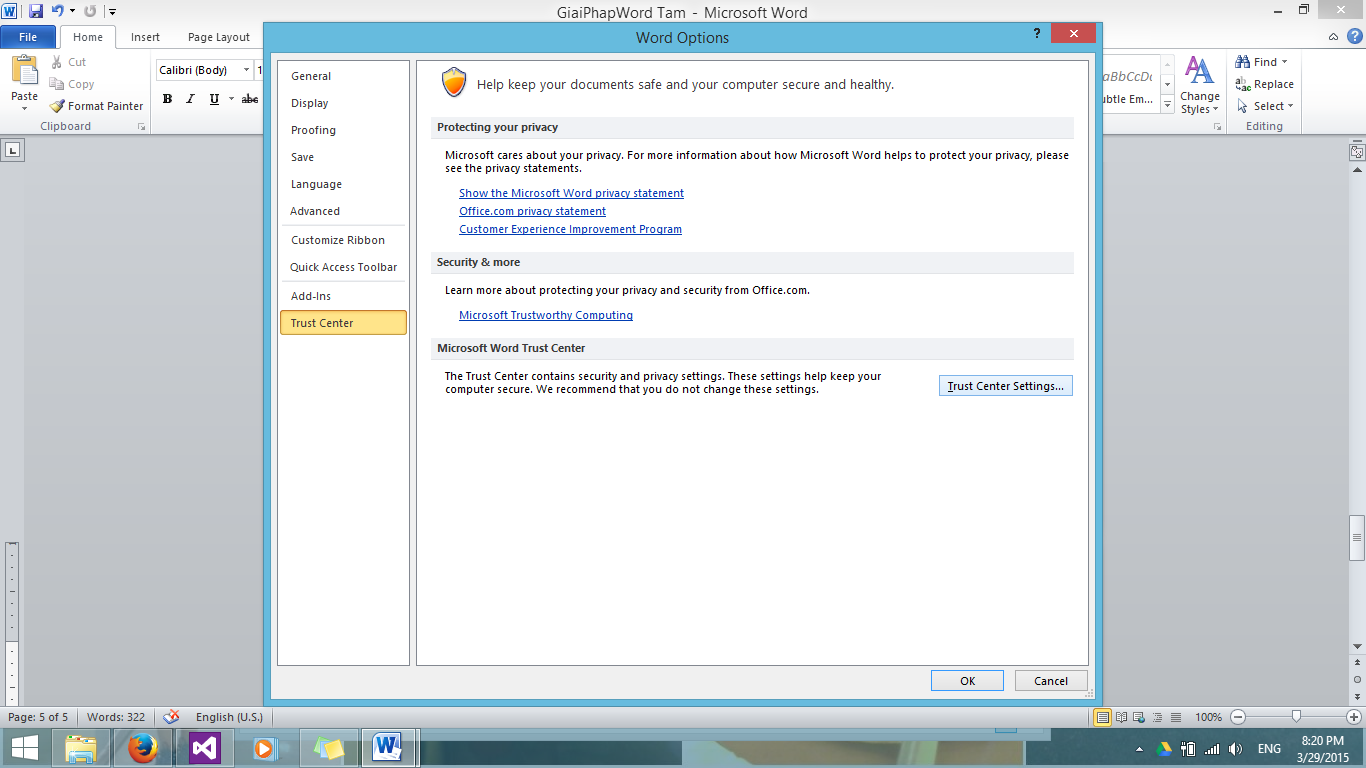


Hình ‑ Cài đặt WordAddin1

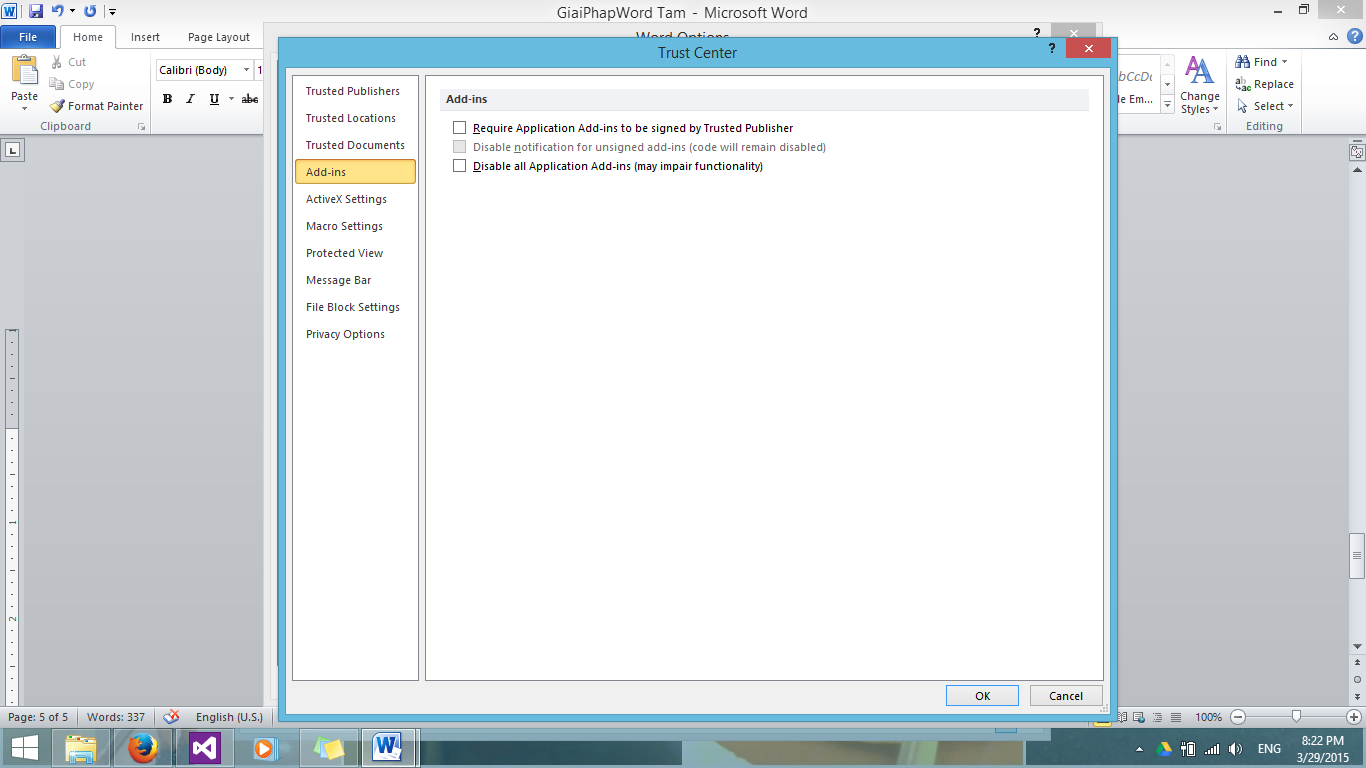
Với các vấn đề về security :

Về việc cài đặt thì vẫn chưa tìm hiểu được do bước cài đặt chưa được xử lý

Về các điều kiện để thực thi thì Microsoft Word có phần gọi là Trust Center quản lý các mức độ tin tưởng ví dụ như yêu cầu add-in phải có chữ ký của Trusted Publisher hoặc tắt hết các add-ins. Hầu như ai cũng có thể vào thay đổi settings nếu họ biết.

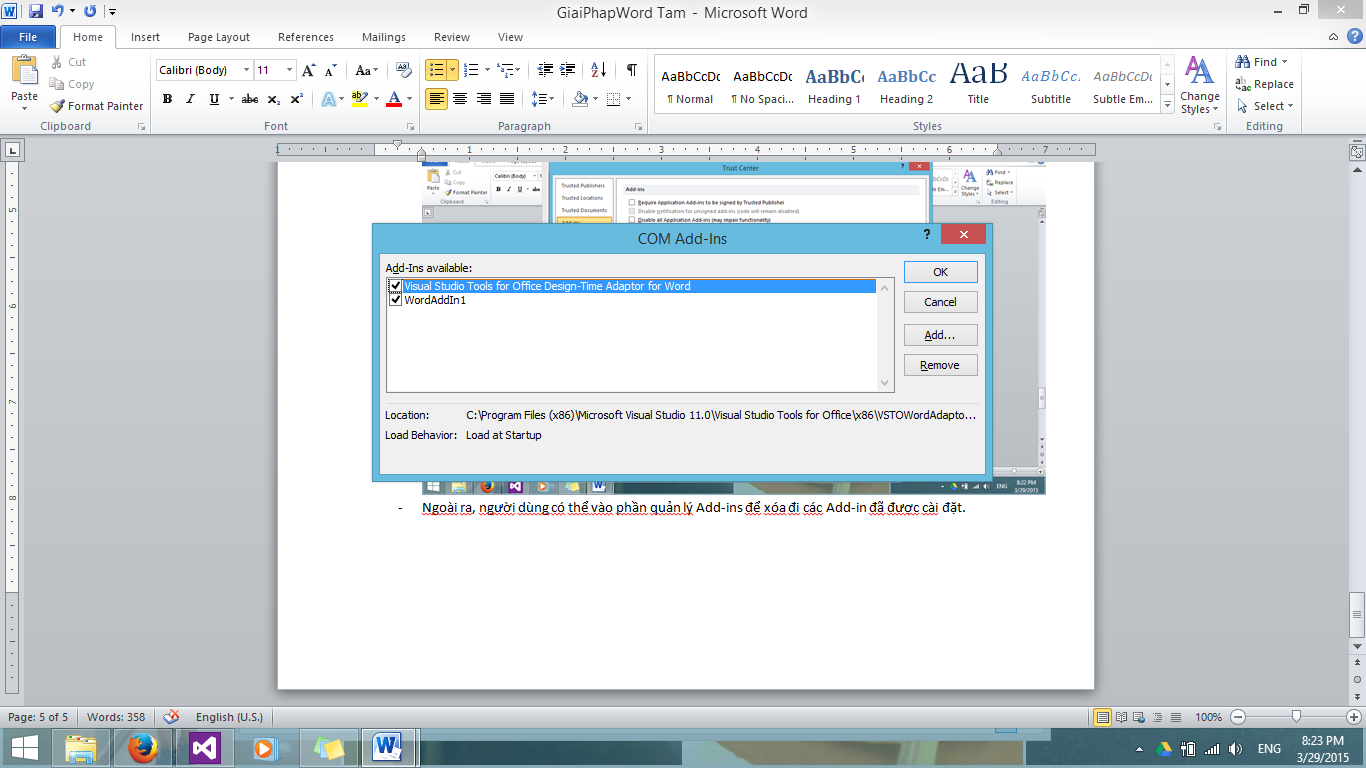


Hình ‑ Word Options – Trust Center



Hình ‑ Trust Center – Add-ins

Ngoài ra, người dùng có thể vào phần quản lý Add-ins để xóa đi các Add-in đã được cài đặt.



Hình ‑ Quản lý Add-ins trong Word

### Tự mã hóa dữ liệu

Giải pháp 1: Mã hóa nội dung text. Sử dụng thuật toán mã hóa TRIPDES. Nội dung sau khi giải mã bị mất format, mất hình ảnh, biểu tượng...

Giải pháp 2: Mã hóa toàn bộ file trước khi đóng ứng dụng. Không được phép khi file vẫn đang mở bằng Word.

# Bảo vệ dữ liệu

* Nội dung của Chương 5**Error! Reference source not found.** trình bày các giải pháp phát hiện thiết bị lưu trữ di động tương tác với máy tính. Từ đó, lấy được thông tin cần thiết cho việc bảo vệ dữ liệu khi sao chép vào thiết bị.

# Phát hiện thiết bị lưu trữ di động tương tác với máy tính

* Nội dung của Chương 5**Error! Reference source not found.** trình bày các giải pháp phát hiện thiết bị lưu trữ di động tương tác với máy tính. Từ đó, lấy được thông tin cần thiết cho việc bảo vệ dữ liệu khi sao chép vào thiết bị.
* Vấn đề

Thiết bị lưu trữ di động cần được bảo vệ dữ liệu trong đó. Trước hết, chúng ta cần biết khi nào thiết bị tương tác với máy tính để tiến hành xử lý dữ liệu được đưa vào.

* Giải pháp

Chúng ta có 2 giải pháp C++ và C# sau đây:

## Giải pháp C++

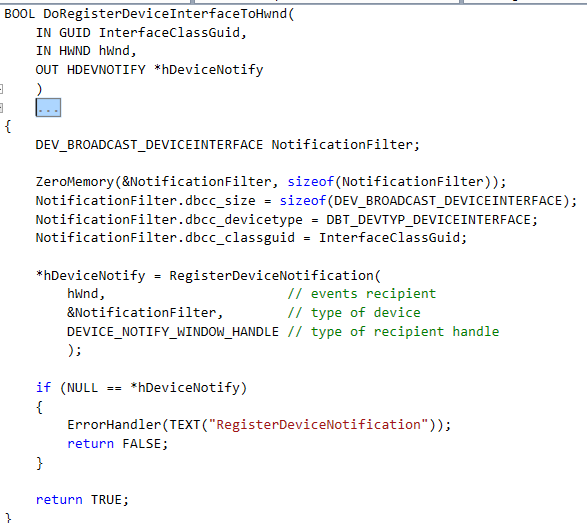
Trong Windows, mỗi ứng dụng sẽ gửi thông báo sự kiện bằng hàm **BroadcastSystemMessage()**. Mọi ứng dụng tầng trên có thể nhận thông báo bằng cách xử lý tin nhắn WM\_DEVICECHANGE. Ứng dụng có thể dùng hàm **RegisterDeviceNotification()** để đăng ký nhận thông báo về các thiết bị.

Service có thể dùng hàm **RegisterDeviceNotification()** để đăng ký nhận thông báo thiết bị. Nếu một service chỉ định một window handle trong **hRecipient** parameter, thông báo được gửi đến window procedure. Nếu hRecipient là một dịch vụ kiểm soát trạng thái, **SERVICE\_CONTROL\_DEVICEEVENT** thông báo được gửi đến cho dịch vụ điều khiển xử lý.

Sự kiện **DBT\_DEVICEARRIVAL** và **DBT\_DEVICEREMOVECOMPLETE** được tự động phát thông báo cho tất cả các ứng dụng tầng trên cho cổng các thiết bị. Hơn thế nữa, việc này không cần thiết gọi RegisterDeviceNotification cho các cổng, và hàm thất bại nếu dbch\_devicetype là **DBT\_DEVTYP\_VOLUME**. Thiết bị OEM-defined không sử dụng trực tiếp bởi hệ thống, do đó cá chức năng thất bại nếu dbch\_devicetype là **DBT\_DEVTYP\_OEM**.

Khai báo **GUID** cho tất cả các **USB serial host PnP drivers** (có thể thay thế bằng bất kỳ GUID hợp lệ khác)





Hàm **DoRegisterDeviceInterfaceToHwnd**(IN GUID InterfaceClassGuid, IN HWND hWnd, OUT HDEVNOTIFY \*hDeviceNotify) có chức năng đăng ký cho một Hwnd để nhận thông báo thay đổi giao tiếp của các thiết bị có GUID được chỉ định ở **WcenusbshGUID**. Bên trong hàm **DoRegisterDeviceInterfaceToHwnd**() gọi hàm:

HDEVNOTIFY WINAPI RegisterDeviceNotification(

\_In\_  HANDLE hRecipient,

\_In\_  LPVOID NotificationFilter,

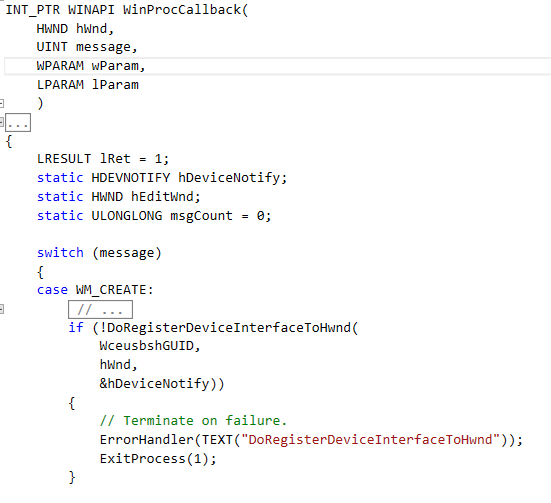
\_In\_  DWORD Flags

);

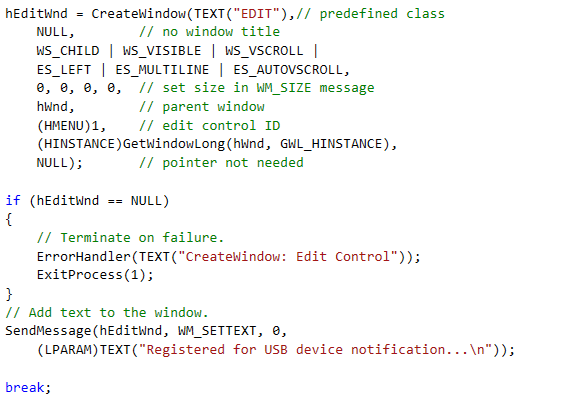
* *hRecipient* [in]: một handle của window hoặc service sẽ nhận các sự kiện của thiết bị được chỉ định trong tham số *NotificationFilter*. Những window handle tương tự có thể sử dụng trong multiple calls tới **RegisterDeviceNotification**. Service có thể xác định cả windows handle và service status handle.
* *NotificationFilter* [in]: Một con trỏ tới block dữ liệu xác định kiểu thiết bị mà thông báo nên được gửi. Block này luôn bắt đầu với cấu trúc **DEV\_BROADCAST\_HDR**. Dữ liệu theo sau header này phụ thuộc vào giá trị của **dbch\_devicetype**, nó có thể là **DBT\_DEVTYP\_DEVICEINTERFACE** hoặc **DBT\_DEVTYP\_HANDLE**.
* *Flags* [in]: Tham số này có thể là một trong các giá trị sau:

|  |  |
| --- | --- |
| **DEVICE\_NOTIFY\_WINDOW\_HANDLE**  0x00000000 | Tham số *hRecipient* là một window handle. |
| **DEVICE\_NOTIFY\_SERVICE\_HANDLE**  0x00000001 | Tham số *hRecipient* là một service status handle. |

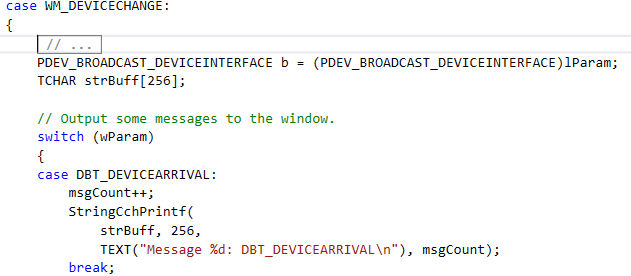
Hàm WindProcCallback(…) tiến hành đăng ký nhận thông báo trong WM\_CREATE. Gọi hàm DoRegisterDeviceInterfaceToHwnd() để đăng ký một lần duy nhất khi khởi tạo



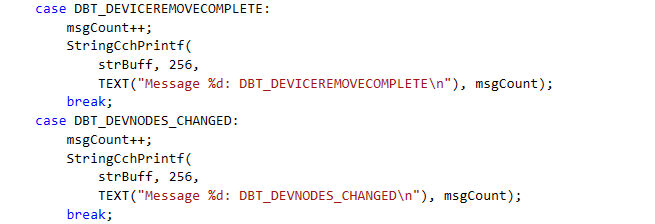
Cuối cùng, tạo cửa sổ giao diện trong WM\_CREATE



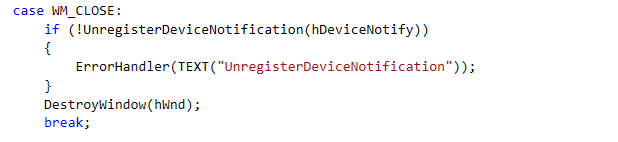
Xử lý trong WM\_DEVICECHANGE



Xuất ra màn hình thông báo khi máy tính nhận thiết bị



Tương tự đối với các thông báo khác nhận được, chương trình xuất ra câu thông báo tương ứng.



Khi đóng ứng dụng, thực hiện thao tác hủy đăng ký nhận thông báo với hàm

BOOL WINAPI UnregisterDeviceNotification(

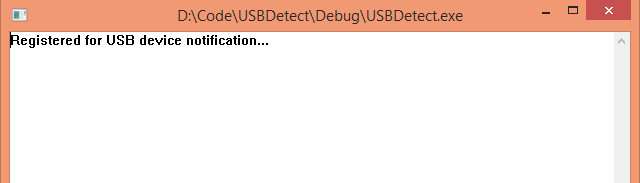
\_In\_  HDEVNOTIFY Handle

);

*Handle* [in]: được trả về khi đăng ký thành công **RegisterDeviceNotification**()

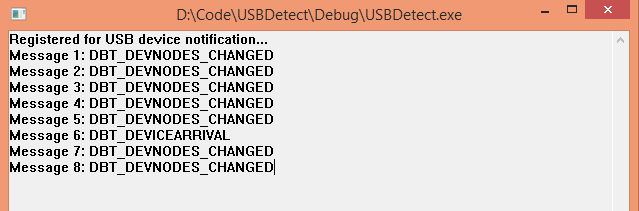
Demo

Giao diện khi khởi động



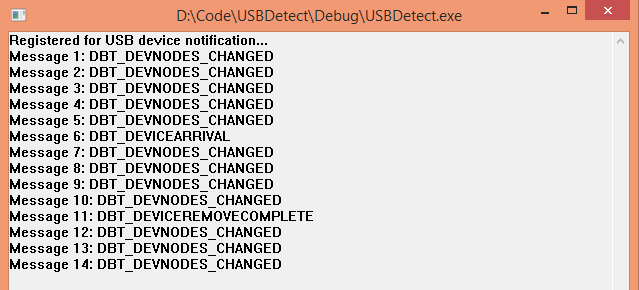
Hình ‑ Giao diện ban đầu

Khi cắm 1 USB vào máy tính



Hình ‑ Chương trình thông báo khi máy tính nhận USB

Khi rút USB ra khỏi máy tính



Hình ‑ Chương trình thông báo khi USB không tương tác

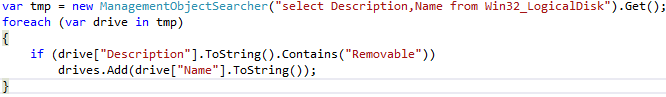
## Giải pháp C#

Sử dụng **WMI** để phát hiện các sự kiện **insert**/**remove** USB trên máy tinh hay laptop.

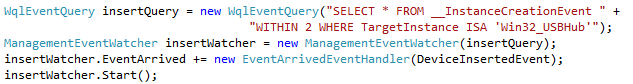
**Windows Management Instrumentation** (WMI) là việc cài đặt của Microsoft Web-Based Management Enterprise (WBEM), nó là một sáng kiến công nghiệp để phát triển một chuẩn công nghệ cho việc truy cập thông tin quản lý trong môi trường doanh nghiệp. WMI sử dụng **Common Information Model** (CIM) tiêu chuẩn công nghiệp để đại diện cho các hệ thống, các ứng dụng, mạng lưới, thiết bị, quản lý và các thành phần khác. CIM được phát triển và duy trì bởi Management Task Force phân tán (DMTF).

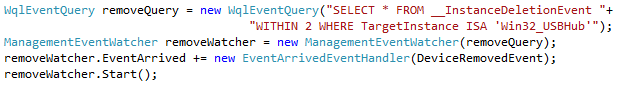
Windows Management Instrumentation (WMI) is the Microsoft implementation of Web-Based Enterprise Management (WBEM), which is an industry initiative to develop a standard technology for accessing management information in an enterprise environment. WMI uses the Common Information Model (CIM) industry standard to represent systems, applications, networks, devices, and other managed components. CIM is developed and maintained by the Distributed Management Task Force (DMTF).

Khi khởi động chương trình thì ta cũng đã khởi tạo tìm kiếm 1 list chuỗi chứa các ký tự tên của các USB hiện đang có trong máy để phục vụ cho công việc xác định được USB nào bị remove.

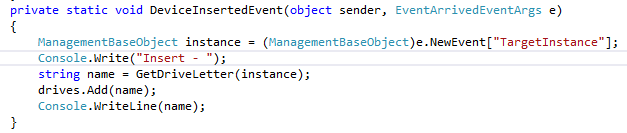


Sau đó ta bắt đầu cho chương trình theo dõi các sự kiện InsertUSB và RemoveUSB





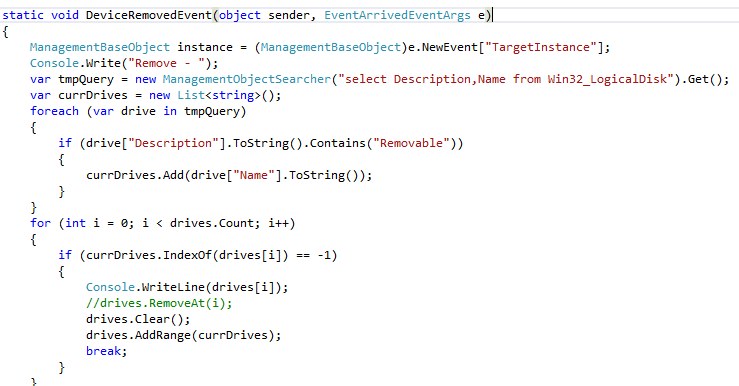
Khi các sự kiện xảy ra thì các phương thức tương ứng sẽ được thực hiện. Với sự kiện insert thì ta sẽ có phương **DeviceInsertedEvent**



Ở trong phương **DeviceInsertedEvent** ta gọi đến phương **GetDriveLetter** để lấy được ký tự của USB khi được insert vào máy và thêm ký tự đó vào list chuỗi mà ta đã khởi tạo lúc đầu. Ta cũng sử dụng các class phương thức, SQL do WMI cung cấp để tìm kiếm ký tự được cấp cho USB.

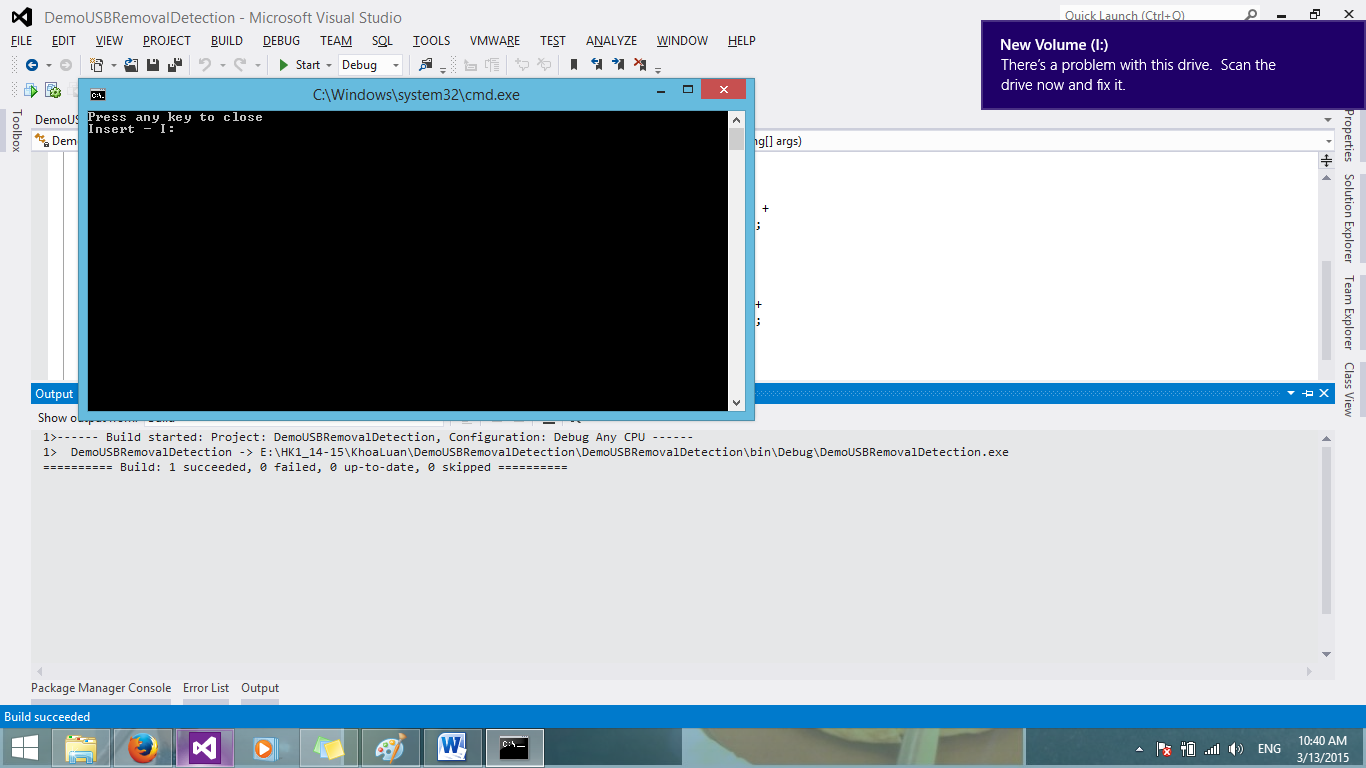


Tương tự cho sự kiện Remove USB ta có phương thức **DeviceRemovedEvent**



Trong đoạn code xử lý khi remove thiết bị thì điều mà ta muốn làm chính là xác định được ký tự được cấp cho USB đã remove là gì.

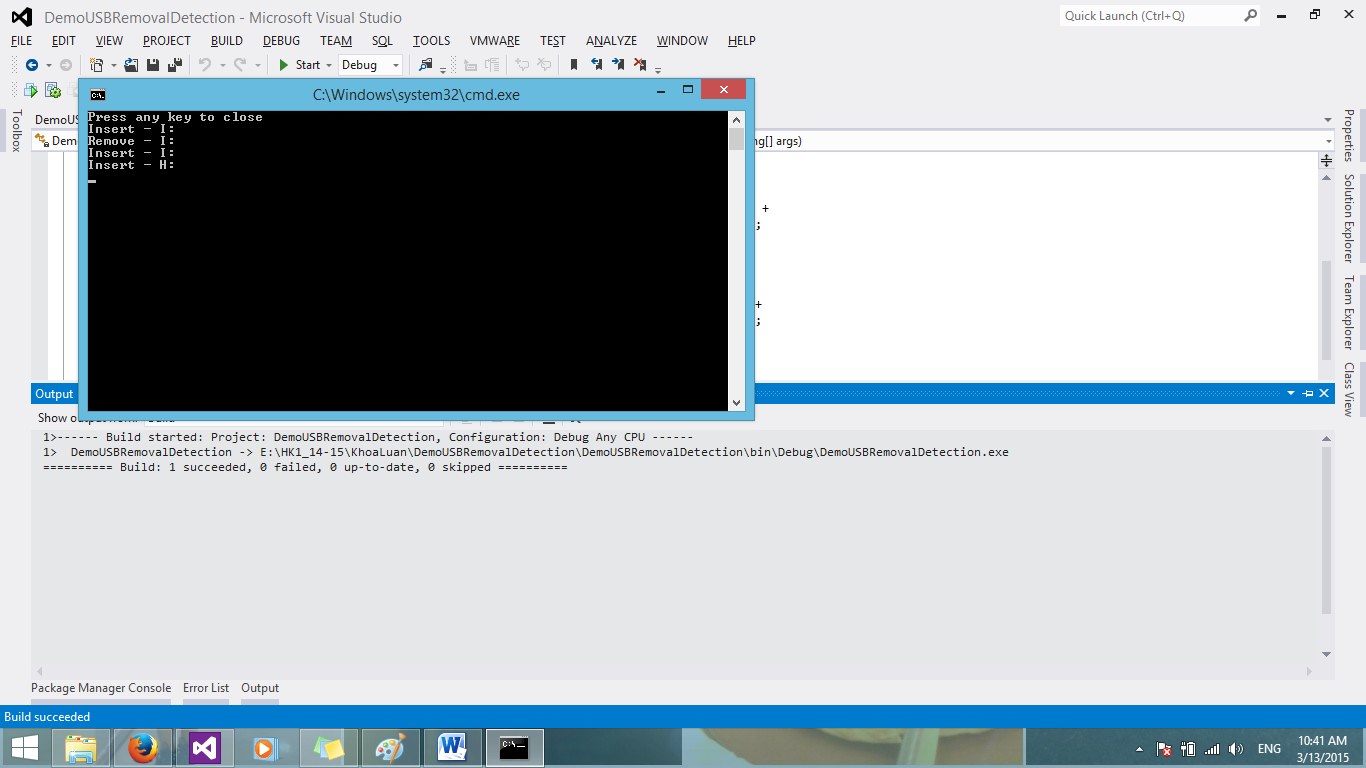
Dưới đây là hình ảnh Demo



Hình ‑ Chương trình thông báo khi có USB tương tác



Hình ‑ Chương trình thông báo khi USB không còn tương tác



Hình ‑ Thay thế một USB khác

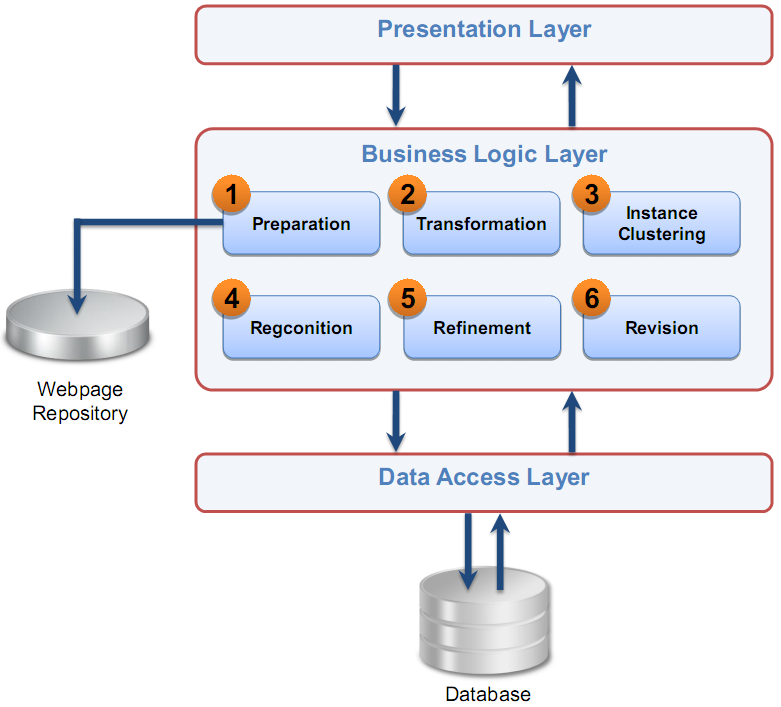
## Kết luận

Nội dung Chương 5 đã trình bày 2 giáp pháp phát hiện USB tương tác với máy tính.

# Kiến trúc hệ thống

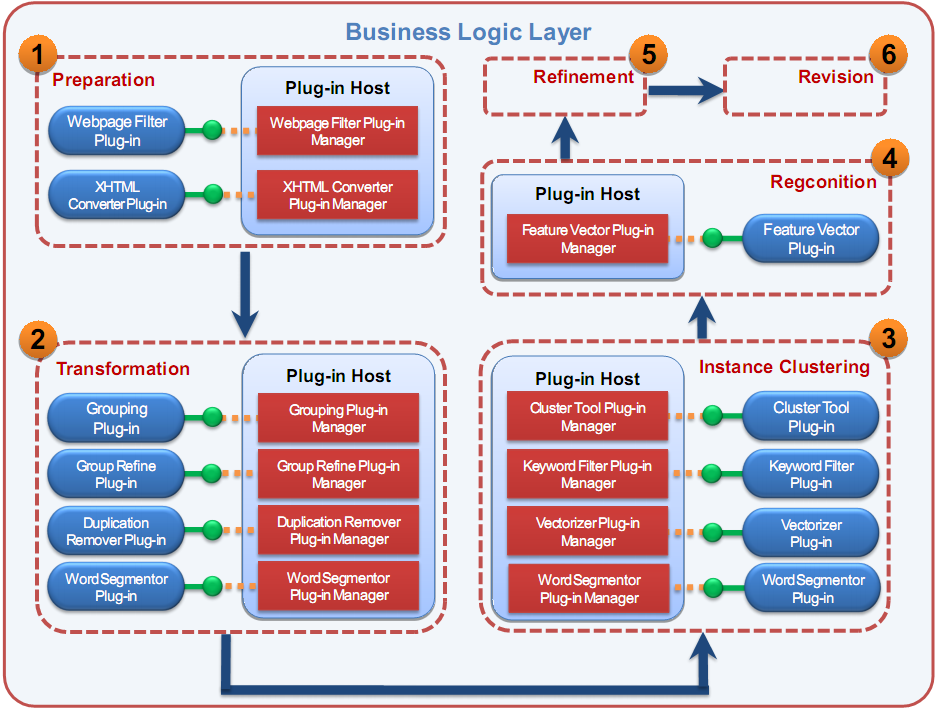
* Nội dung **Error! Reference source not found.** trình bày hệ thống framework được nhóm phát triển để hiện thực hóa quy trình rút trích ontology từ WWW đã được trình bày ở **Error! Reference source not found.**. Các vấn đề và giải pháp khi xây dựng framework và hiện thực hóa quy trình cũng được trình bày trong chương này.

## Kiến trúc hệ thống



Hình ‑ Kiến trúc tổng quát Ontology Extractor Framework

Hình 6‑1 trình bày kiến trúc tổng quát của framework rút trích ontology từ WWW. Hệ thống được thiết kế theo mô hình 3 lớp, trong đó các phân hệ xử lý chính nằm ở tầng giữa gồm 6 phân hệ: Phân hệ Chuẩn bị (Preparation), Phân hệ Chuyển đổi (Transformation), Phân hệ Gom cụm (Instance Clustering), Phân hệ Nhận diện (Regconition), Phân hệ Liên kết (Refinement) và Phân hệ Tinh chỉnh (Revision). Kiến trúc chi tiết của các phân hệ chính được trình bày ở Hình 6‑2.



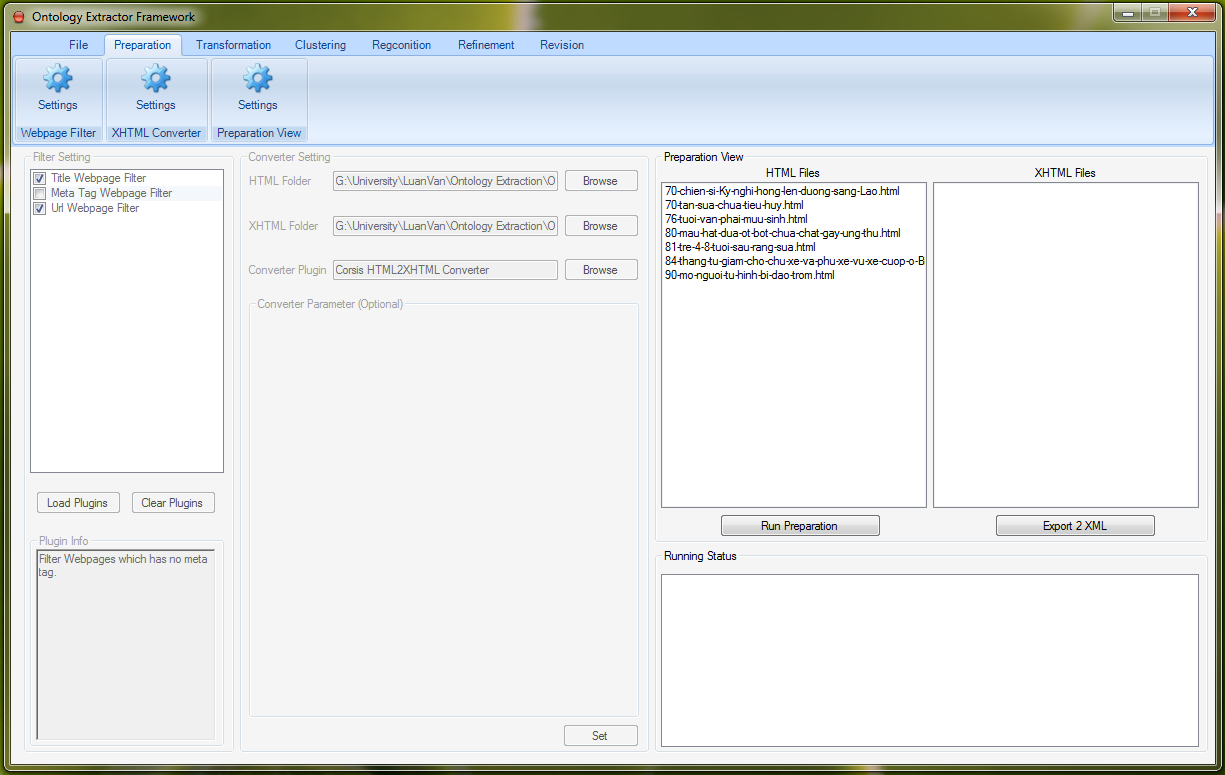
Hình ‑ Kiến trúc các phân hệ chính của Ontology Extractor Framework

Hình 6‑2 trình bày kiến trúc các phân hệ chính của framework rút trích ontology từ WWW. Trong đó:

1. Phân hệ Chuẩn bị (Preparation): hiện thực hóa bước *Chuẩn bị* nhằm xử lý các trang web làm dữ liệu đầu vào cho toàn bộ hệ thống. Trong đó thành phần chính sử dụng kiến trúc Plug-in hỗ trợ khả năng tùy biến cho 2 tác vụ chính của bước này là loại bỏ trang web không phù hợp và chuẩn hóa trang web.
2. Phân hệ Biến đổi (Transformation): hiện thực hóa bước *Biến đổi* với các tác vụ là gom nhóm các trang web theo cấu trúc, loại bỏ các trang web dư thừa và các thành phần lặp. Các tác vụ chính này được quản lý bằng cách sử dụng kiến trúc Plug-in hỗ trợ khả năng tùy biến cao. Ngoài ra, trong phân hệ này còn hỗ trợ tác vụ lưu trữ các trang web vào cơ sở dữ liệu quan hệ (mục 6.3.6).
3. Phân hệ Gom cụm (Instance Clustering): hiện thực hóa bước *Gom cụm* có nhiệm vụ chính là gom cụm các trang web dựa trên độ tương đồng giữa các *vector đặc trưng* của trang web. Phân hệ này cũng sử dụng kiến trúc Plug-in quản lý các tác vụ chính bao gồm chọn *từ khóa* làm chiều của các *vector đặc trưng*, xây dựng *vector đặc trưng* và gom cụm.
4. Phân hệ Nhận diện (Regconition): hiện thực hóa bước *Nhận diện* có hỗ trợ khả năng tùy biến trong việc rút trích đặc trưng cụm sử dụng kiến trúc Plug-in.
5. Phân hệ Liên kết (Refinement): hiện thực hóa bước *Liên kết* nhằm thực hiện tác vụ xác định *mối quan hệ* giữa các *khái niệm* - được đại diện bằng một cụm.
6. Phân hệ Tinh chỉnh (Revision): hiện thực hóa bước *Tinh chỉnh* nhằm hỗ trợ chuyên gia ontology trong việc lựa chọn các *từ khóa ứng viên* của các *khái niệm*.

Chi tiết về kiến trúc cũng như các vấn đề và giải pháp trong quá trình hiện thực hóa của từng phân hệ được chúng em trình bày trong các phần tiếp sau đây.

## Phân hệ Chuẩn bị (Preparation)



Hình ‑ Giao diện Phân hệ Chuẩn bị

### Kiến trúc phân hệ

Phân hệ Chuẩn bị hiện thực hóa bước đầu tiên trong quy trình 6 bước nhằm xử lý các trang web đầu vào cho toàn bộ hệ thống. Các trang web này cần được xử lý sao cho thỏa những yêu cầu như phải cùng thuộc về một Website, chỉ chứa văn bản, được lưu trữ dưới dạng chuẩn hoá và được gán nhãn từ khoá. Riêng việc gán nhãn từ khoá các trang web, nguồn từ khoá được lựa chọn từ thẻ meta hoặc tách từ tiêu đề của trang web. Các từ khoá này bám sát nội dung thông tin trang web hơn hẳn so với phần văn bản của các đường dẫn đến trang web vốn thể hiện những nội dung thông tin không liên quan. Đây cũng chính là một trong những điểm khác biệt giữa việc hiện thực hóa quy trình của chúng em và nội dung lý thuyết được các tác giả đề nghị. Kiến trúc của phân hệ sử dụng cơ chế plug-in được trình bày ở Hình 6‑4.



Hình ‑ Kiến trúc Phân hệ Chuẩn bị

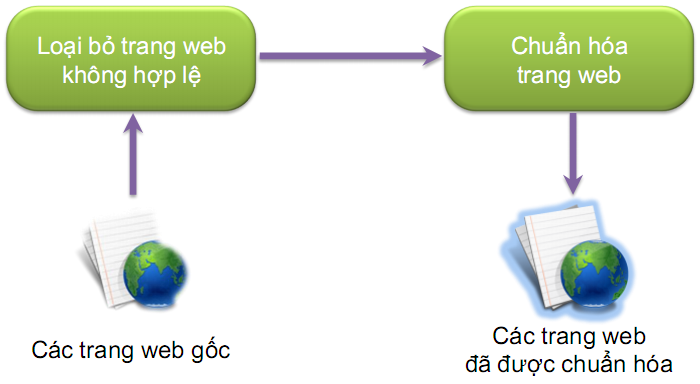
Hình 6‑4 mô tả kiến trúc của phân hệ Chuẩn bị. Phân hệ gồm một host quản lý các plug-in và có 2 loại plug-in. Trong đó chúng em đã xây dựng các thể hiện cho mỗi loại plugin:

* Webpage Filter Plug-in: đóng vai trò là bộ lọc trang web không thỏa yêu cầu. (3 thể hiện)
* XHTML Converter Plug-in: là công cụ chẩn hóa trang web. (1 thể hiện)

Bảng 6‑1 Các lớp chính cấu thành phân hệ Chuẩn bị trình bày các lớp đối tượng chính cấu thành phân hệ Chuẩn bị.

|  |  |  |
| --- | --- | --- |
| STT | Tên Lớp | Mô tả |
| 1 | IPreparationHost | Interface mô tả host quản lý các plug-in trong phân hệ Chuẩn bị |
| 2 | PreparationHost | Thể hiện host của phân hệ Chuẩn bị |
| 3 | IwebpageFilter | Interface mô tả plug-in bộ lọc trang web |
| 4 | MetaWebpageFilterPlugin | Bộ lọc các trang web không chứa meta data |
| 5 | UrlWebpageFilterPlugin | Bộ lọc các trang web không chứa đường dẫn gốc |
| 6 | TitleWebpageFilterPlugin | Bộ lọc các trang web không chứa tiêu đề |
| 7 | IXHTMLConverter | Interface mô tả plug-in chuẩn hóa trang web |
| 8 | CorsisXHTMLConverterPlugin | Plug-in chuẩn hóa trang web sử dụng thư viện Corsis chuyển đổi sang định dạng XHTML |

Bảng 6‑1 Các lớp chính cấu thành phân hệ Chuẩn bị



Hình ‑ Mô hình xử lý tổng quát Phân hệ Chuẩn bị

Hình 6‑5 mô tả quy trình xử lý tổng quát của phân hệ Chuẩn bị. Đầu tiên, các trang web gốc được đưa qua bộ lọc để loại bỏ những trang web không phù hợp như trang web không có meta data hoặc không có tiêu đề (6.2.3). Sau đó, chúng được chuẩn hóa bằng cách chuyển đổi định dạng sang XHTML (6.2.4). Kết quả của phân hệ này là tập các trang web thỏa yêu cầu của người sử dụng và đã được chuẩn hóa.

### Tải các trang web về lưu trữ ngoại tuyến

* Vấn đề:

Do mục đích của hệ thống là rút trích ontology từ một *Website mục tiêu*, nên đối tượng xử lý chính của phương pháp là các trang web của Website đó. Mặt khác, số lượng các trang web là rất lớn: việc xử lý ở đây mang tính chất hàng loạt, nghĩa là toàn bộ các trang web phải luôn sẵn sàng cho việc xử lý. Do đó, để đảm bảo tính liên tục, không bị ngắt quãng của hệ thống, từ đó phát sinh nhu cầu tải toàn bộ các trang web thuộc về cùng một Website để lưu trữ ngoại tuyến. Việc tải các trang web này tạo ra tập các trang web gốc được phân hệ Chuẩn bị sử dụng như bộ dữ liệu đầu vào.

* Giải pháp:

Giải pháp được đề ra là sử dụng một công cụ crawler nhằm tải toàn bộ các trang web của Website về lưu trữ trên máy tính. Hiện nay, có rất nhiều công cụ crawler như: OfflineExplorer, Teleport, HTTrack....Có 2 hướng tiếp cận cho việc sử dụng công cụ crawler như sau:

* Sử dụng công cụ crawler mã nguồn mở: có thể sử dụng các chương trình mã nguồn mở có sẵn và chỉnh sửa, bổ sung thêm những chức năng cần thiết phù hợp với đặc thù công việc.
* Sử dụng công cụ crawler có bản quyền: với các phần mềm này, không thể chỉnh sửa mã nguồn mà chỉ có thể sử dụng các tùy chọn có sẵn của chương trình nhằm chuẩn bị một cách thuận lợi nhất cho công việc.

Lựa chọn của nhóm là sử dụng Offline Explorer (OE) – một crawler có bản quyền của Metaproducts. OE có những ưu điểm sau:

* Tùy chọn loại tập tin, nội dung tập tin được tải: ởđây, tùy chọn được quan tâm là *chỉ tải văn bản (text)* của các trang web.
* Multi Thread: OE hỗ trợ tải nhiều trang web cùng một lúc, điều này giúp tăng tốc độ thu thập dữ liệu cho hệ thống.
* Tùy chọn Server: OE hỗ trợ chỉ tải những trang web có đường dẫn (URL) thuộc về cùng một server. Tùy chọn này rất phù hợp với mục đích của đề tài, đó là rút trích ontology từ những trang web thuộc cùng một *lĩnh vực*.
* Lưu giữ được đường dẫn (URL) gốc: một nhu cầu thiết yếu khác đó là phải lưu giữ được đường dẫn gốc của các trang web. Đường dẫn gốc hỗ trợ hệ thống phân loại trang web được hiệu quả hơn.
* Tạo ra một phiên bản*“phẳng”*của Website: OE hỗ trợ kết xuất tất cả các trang web vào một thư mục đồng thời chỉnh sửa tất cả các đường dẫn (href) nhằm đảm bảo mối liên kết giữa tất cả các trang web như ban đầu. Điều này tạo thuận lợi cho quá trình xử lý và duyệt qua các trang web.

Phiên bản được chúng em sử dụng trong cài đặt là Offline Explorer Enterprise 5.4.2958.

### Loại bỏ trang web không hợp lệ

* Vấn đề

Do đặc thù của công cụ crawler, một số trang web không phù hợp vẫn được tải về như các trang web thuộc Website khác được gọi gián tiếp từ *Website mục tiêu*, các trang web không chứa dữ liệu meta hoặc các trang web không thể lưu trữ được đường dẫn gốc.

Mặt khác, số lượng trang web được lưu trữ và xử lý là rất lớn, vì vậy hệ thống cần được tối ưu hoá về thời gian và không gian lưu trữ. Chính vì vậy, hệ thống cần lọc bỏ những trang web không phù hợp nhằm giảm số lượng trang web sẽ được xử lý ở những bước tiếp theo cũng như loại bỏ những trang web không thỏa mãn yêu cầu xử lý của các tác vụ thuộc những phân hệ tiếp theo.

* Giải pháp

Việc sử dụng các công cụ crawler khác nhau sẽ phát sinh những trang web không phù hợp khác nhau đối với nhóm những Website khác nhau, vì thế hệ thống phải có khả năng hỗ trợ người dùng loại bỏ những trang web không phù hợp theo những tiêu chí khác nhau dành cho các nhóm Website khác nhau.

Giải pháp được đề ra là xây dựng một bộ lọc dữ liệu hỗ trợ các tuỳ chọn có sẵn để người dùng có thể loại bỏ những trang web không phù hợp. Ngoài ra, bộ lọc này còn có khả năng mở rộng và phát triển thêm các tuỳ chọn mới sau này. Vì vậy, kiến trúc Plug-in được sử dụng nhằm hỗ trợ khả năng tùy biến trong việc lựa chọn các bộ lọc. Bảng sau đây trình bày các bộ lọc mà chúng em đã xây dựng và triển khai.

|  |  |  |
| --- | --- | --- |
| STT | Bộ lọc trang web | Mô tả |
| 1 | MetaWebpageFilterPlugin | Bộ lọc các trang web không chứa meta data |
| 2 | UrlWebpageFilterPlugin | Bộ lọc các trang web không chứa đường dẫn gốc |
| 3 | TitleWebpageFilterPlugin | Bộ lọc các trang web không chứa tiêu đề |

Bảng 6‑2 Các bộ lọc trang web

### Chuẩn hoá các trang web

* Vấn đề:

Phương pháp của đề tài tập trung vào việc phân tích, xử lý nội dung các trang web dựa vào cấu trúc các thẻ. Phần lớn các Website Việt Nam hiện nay đều được viết bằng HTML không chuẩn hóa, vì vậy gây rất nhiều khó khăn hoặc thậm chí ảnh hưởng đến chất lượng đầu ra của hệ thống.

* Giải pháp:

Một trong những ngôn ngữ đánh dấu (Markup Language) được tổ chức W3C khuyến khích các nhà phát triển Website sử dụng là XHTML. Kết hợp những ưu điểm vượt trội của XML vào HTML, XHTML ra đời. XHTML mang những tính chất chuẩn mực của XML (<http://www.w3schools.com/xhtml/default.asp>) như:

* Các thẻ XHTML phải được lồng nhau đúng cách.
* Các thẻ XHTML mở phải luôn có thẻ đóng đi kèm.
* Tên các thẻ XHTML phải được viết bằng chữ thường.
* Mỗi văn bản XHTML phải có một phần tử gốc duy nhất.

Vì thế, nhu cầu của chúng em chính là chuyển đổi các trang web nguồn được viết bằng HTML thành XHTML chuẩn; điều này sẽ giúp ích cho quá trình xử lý và phân tích sau này. Bước chuyển đổi phải tuân thủ những yêu cầu như sau:

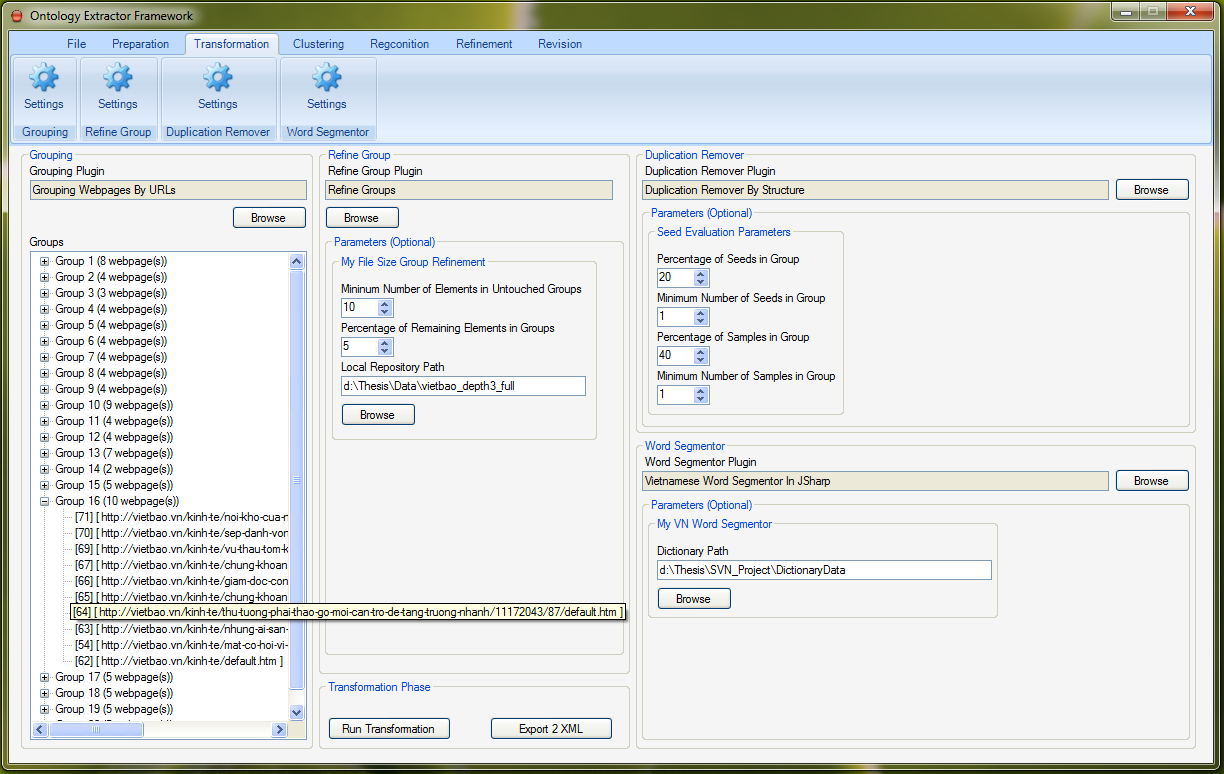
* Cách sửa lỗi các trang web phải giữ lại và đảm bảo được *mối tương quan đúng* giữa các thẻ.
* Các trang web có lỗi giống nhau phải được chuyển đổi giống nhau. Như vậy, các trang có cấu trúc thẻ giống nhau sau bước chuyển đổi vẫn giữ nguyên được mối liên quan về cấu trúc này.

Ngoài ra, công nghệ phát triển Website ngày nay là xây dựng hàng loạt các trang web theo một khuôn mẫu (template) hoặc bộ khung (layout) có sẵn. Như vậy, các lỗi định dạng HTML thiếu chuẩn mực thường mang tính *hàng loạt*. Từ nhận định trên, việc nhóm thực hiện việc chuyển đổi từ HTML sang XHTML cho tất cả các trang web có được mà vẫn đảm bảo sự tương quan cấu trúc các thẻ ban đầu.

Hiện nay có khá nhiều công cụ chuyển đổi định dạng HTML sang XHTML,và công cụ opensource Corsis (<http://sourceforge.net/projects/corsis/>) được chúng em lựa chọn để thực hiện việc chuyển đổi định dạng các trang web từ HTML sang XHTML chuẩn hoá.

Phân hệ Chuẩn bị có hỗ trợ tùy biến plug-in cho công cụ chuẩn hóa các trang web. Để có thể sử dụng công cụ Corsis thực hiện tác vụ chuẩn hóa trang web, chúng em xây dựng plug-in CorsisXHTMLConverterPlugin sử dụng dll của Corsis.

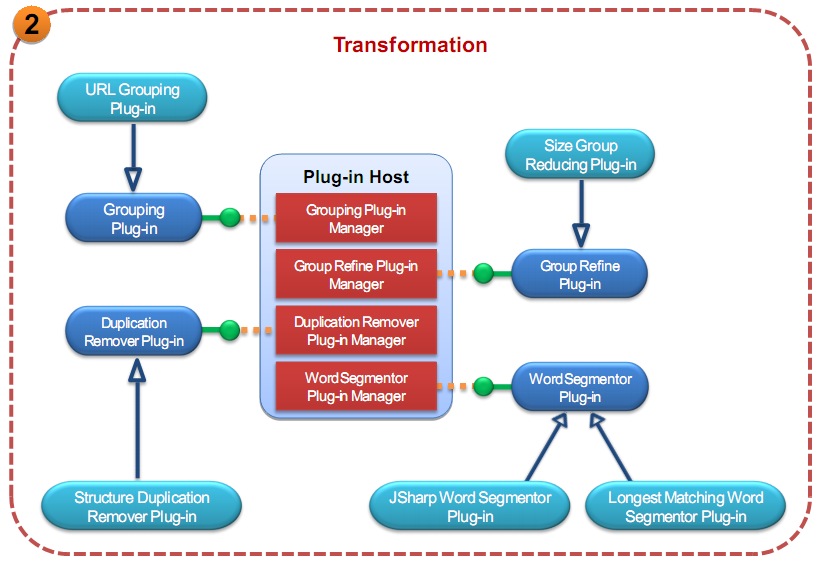
## Phân hệ Biến đổi (Transformation)



Hình ‑ Giao diện phân hệ Biến đổi

### Kiến trúc phân hệ

Các *thành phần lặp* là các phần được lặp đi lặp lại qua các trang web như quảng cáo, thông tin bản quyền Website hay thanh danh mục Website…Các *thành phần lặp* này không mang đặc trưng riêng của mỗi trang web, trong khi mục đích của phương pháp là rút trích ontology từ việc phân tích đặc trưng của các trang web, vì vậy các *thành phần lặp* này cần được loại bỏ trước khi tiến hành các bước phân tích và thống kê tiếp theo. Các *thành phần lặp* thường lặp lại ở các vị trí giống nhau giữa các trang web trong cùng một nhóm có cùng cấu trúc HTML. Và việc loại bỏ các *thành phần lặp* chính là nhiệm vụ chính của phân hệ này. Kiến trúc của phân hệ sử dụng cơ chế plug-in được trình bày ở Hình 6‑7.



Hình ‑ Kiến trúc Phân hệ Biến đổi

Hình 6‑7 mô tả kiến trúc của phân hệ Chuẩn bị. Phân hệ gồm một host quản lý các plug-in và có 4 loại plug-in. Trong đó chúng em đã xây dựng các thể hiện cho mỗi loại plugin:

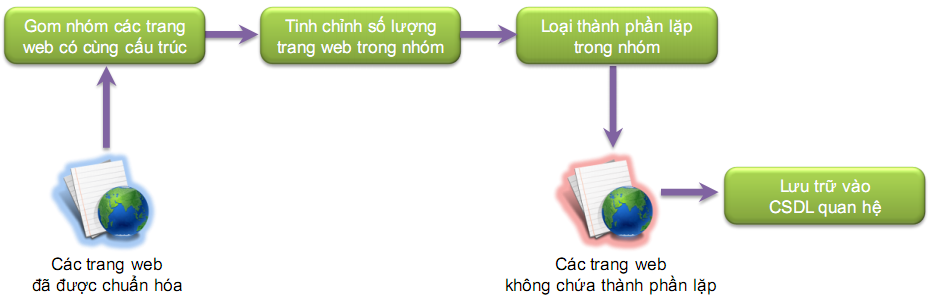
* Grouping Plug-in: gom nhóm các trang web tương đồng về cấu trúc.  
  (1 thể hiện)
* Group Refine Plug-in: tinh chỉnh số lượng trang web của nhóm.  
  (1 thể hiện)
* Duplication Remover Plug-in: loại trừ thành phần lặp giữa các trang web trong cùng một nhóm. (1 thể hiện)
* Word Segmentor Plug-in: tách từ tiếng Việt. (2 thể hiện)

Bảng 6‑3 Các lớp chính cấu thành phân hệ Biến đổi trình bày các lớp đối tượng chính cấu thành phân hệ Biến đổi.

|  |  |  |
| --- | --- | --- |
| STT | Tên Lớp | Mô tả |
| 1 | ItransformationHost | Interface mô tả host quản lý các plug-in trong phân hệ Biến đổi |
| 2 | TransformationHost | Thể hiện host của phân hệ Biến đổi |
| 3 | IwebpageGrouping | Interface mô tả plug-in gom nhóm các trang web |
| 4 | UrlGroupingPlugin | Plugin gom nhóm các trang web dựa trên đường dẫn gốc |
| 5 | IgroupReducing | Interface mô tả plug-in tinh chỉnh số lượng nhóm |
| 6 | SizeGroupReducingPlugin | Plugin tinh chỉnh số lượng nhóm dựa trên kích thước trang web |
| 7 | IduplicationRemover | Interface mô tả plug-in loại thành phần lặp |
| 8 | StructureDuplicationRemoverPlugin | Plugin loại thành phần lặp dựa trên sự tương đồng về cấu trúc trang web |
| 9 | IwordSegmentor | Interface mô tả plug-in tách từ tiếng Việt |
| 10 | JsharpWordSegmentorPlugin | Plug-in tách từ tiếng Việt sử dụng bộ thư viện Jsharp |
| 11 | LongestMatchingWordSegmentorPlugin | Plug-in tách từ tiếng Việt bằng phương pháp so khớp từ dài nhất |

Bảng 6‑3 Các lớp chính cấu thành phân hệ Biến đổi

Hình 6‑8 mô tả quy trình bước *chuẩn bị*, đầu tiên các trang web được gom nhóm dựa sự giống nhau về cấu trúc (6.3.2). Các nhóm này sau đó được tinh chỉnh số lượng các trang web trong nhóm (6.3.3). Kế tiếp, lần lượt các nhóm được loại bỏ *thành phần lặp* giữa các trang web trong nhóm (6.3.4 và 6.3.5). Cuối cùng, các trang web được lưu trữ vào cơ sở dữ liệu quan hệ (6.3.6).



Hình ‑ Mô hình xử lý tổng quát Phân hệ Biến đổi

### Gom nhóm các trang web dựa trên đường dẫn gốc

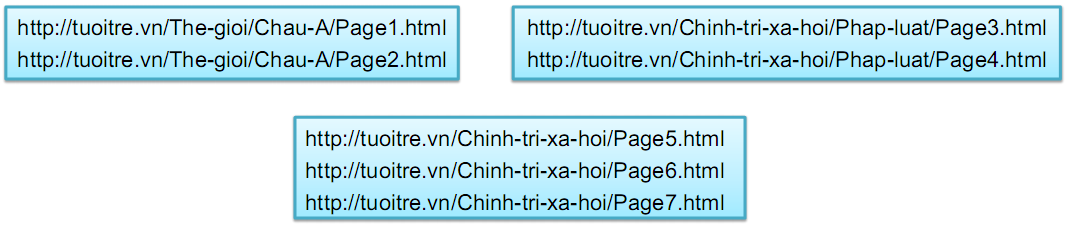
* Vấn đề

Phương pháp gom nhóm các trang web sử dụng *chuỗi đường dẫn* do các tác giả đề nghị đã được phân tích và chỉ ra những hạn chế ở mục **Error! Reference source not found.**, từ đó chúng em đề ra phương pháp gom nhóm dựa trên đường dẫn (URL) của trang web. Đường dẫn phản ánh sự lưu trữ theo phân loại nội dung, chuyên mục của các trang web trên máy chủ. Mặt khác, các trang web thuộc về cùng một chuyên mục thường được tạo ra bởi cùng một cấu trúc khuôn mẫu (template) cho nên sẽ có nhiều sự tương đồng về cấu trúc HTML. Ý tưởng phân nhóm các trang web dựa trên đường dẫn được xây dựng trên nền tảng những nhận xét và phân tích trên.

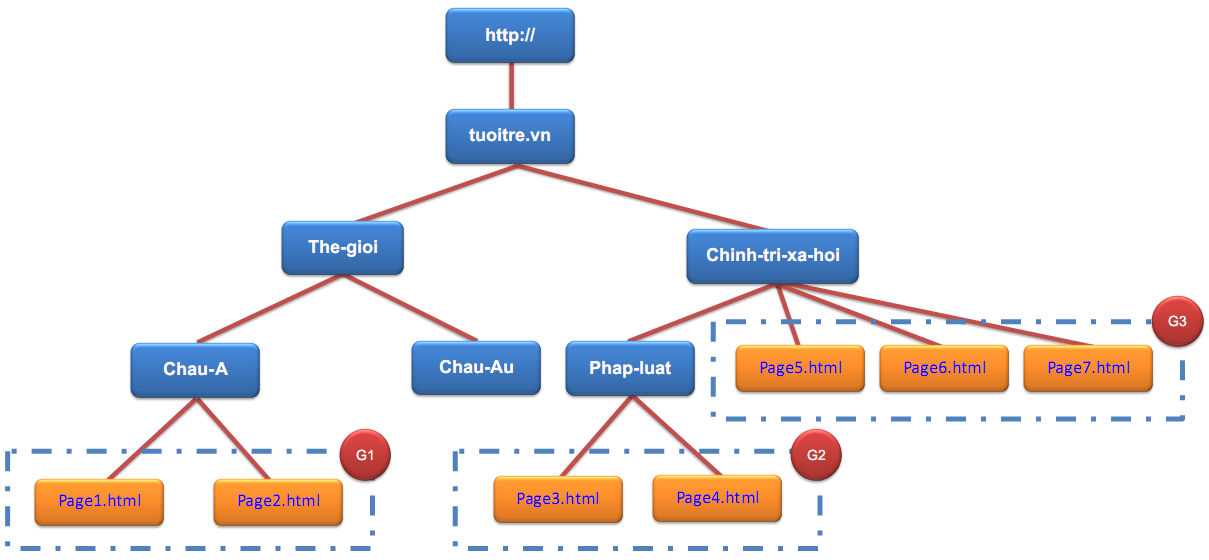
* Giải pháp

Chúng em sử dụng các thành phần của đường dẫn được phân tách bởi các dấu “**/**” để xây dựng nên cây đường dẫn như Hình 6‑10. Trong đó, mỗi một thành phần đường dẫn được xem như là một node trên cây và lá là tên các trang web. Theo đó, các trang web được đại diện bằng các node lá trên cây đường dẫn.

Dễ thấy, các trang web có phần đường dẫn giống nhau sẽ có phần tên thuộc về cùng một node cha. Như vậy, các trang web được xem cùng thuộc một nhóm khi chúng cùng thuộc về một node cha trên *cây đường dẫn*.



Hình ‑ Các đường dẫn gốc được phân nhóm



Hình ‑ Cây đường dẫn

Hình 6‑10 biểu dẫn *cây đường dẫn* xây dựng từ tập 7 đường dẫn ở Hình 6‑9. Trong đó, các trang *Page*1*.html* và *Page*2*.html* cùng thuộc về chuyên mục *Châu Á* trong nhánh *Thế giới* được phân vào *Group* 1. Các trang *Page*5*.html, Page*6*.html* và *Page*7*.html* thuộc về chuyên mục *Chính trị xã hội* nói chung được phân vào *Group* 3. Và các trang *Page*3*.html, Page*4*.html* thuộc về chuyên mục *Pháp luật* được phân thành một nhóm *Group* 2.

Phân hệ Biến đổi hỗ trợ việc tùy biến phương pháp gom nhóm các trang web có cùng cấu trúc. Trong quá trình hiện thực hóa, chúng em đã triển khai plug-in gom nhóm các trang web dựa vào đường dẫn gốc của chúng UrlGroupingPlugin.

### Tinh chỉnh số lượng các trang web trong một nhóm

* Vấn đề

Sau khi phân nhóm các trang web, khả năng tồn tại một số nhóm có số lượng phần tử không cân đối như có quá nhiều hoặc quá ít các trang web thuộc về nhóm. Các nhóm này gây khó khăn cho các bước xử lý tiếp theo. Từ đây xuất phát nhu cầu cần tinh chỉnh các nhóm này sao cho thuận lợi và phù hợp với các xử lý sau đó.

* Giải pháp

Phân hệ này hỗ trợ việc tùy biến phương pháp tinh chỉnh số lượng các trang web trong cùng một nhóm. Trong quá trình hiện thực hóa, chúng em đã triển khai plug-in tinh chỉnh nhóm các trang web SizeGroupReducingPlugin. Trong đó, phương pháp tinh chỉnh nhóm dựa trên đặc thù của phương pháp gom nhóm mà chúng em phát triển:

* Các trang web trong cùng nhóm thuộc về cùng một chuyên mục nên có nội dung tương tự nhau. Vì vậy, phương pháp tinh chỉnh sẽ chỉ giữ lại các trang web có kích thước lớn nhất trong nhóm. Nghĩa là, chỉ các trang web chứa nhiều nội dung thông tin của một chuyên mục sẽ được giữ lại.
* Các nhóm chỉ có một phần tử sẽ được gộp với nhóm gần nó nhất. Vì vậy, không tồn tại các nhóm có một phần tử. Do đó, các phương pháp tinh chỉnh sẽ bỏ qua các nhóm có số phần tử ít nhất là 2. Vì với một nhóm có 2 phần tử, các *thành phần lặp* vẫn có thể được loại trừ.

### Loại trừ các thành phần lặp giữa 2 trang web

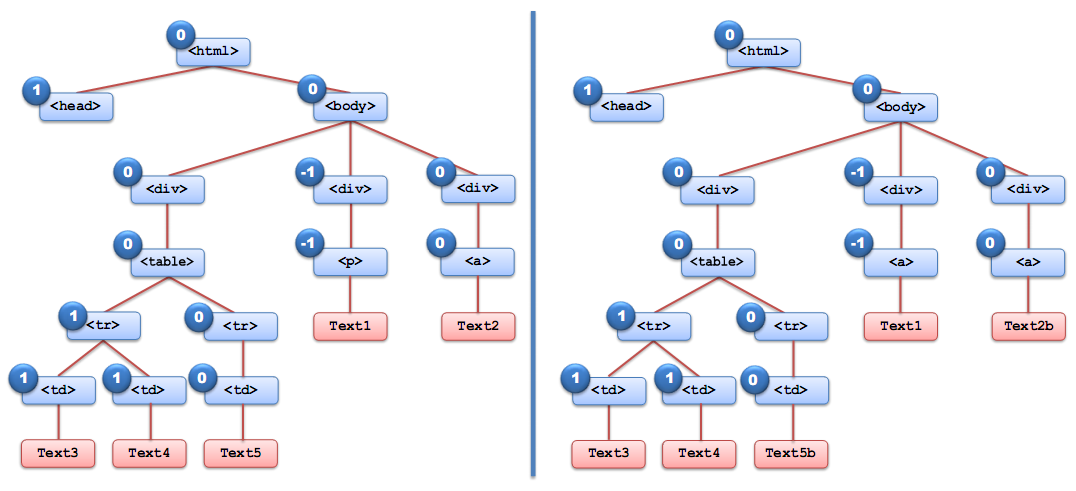
* Vấn đề:

Như đã được phân tích ở mục **Error! Reference source not found.**, phương pháp so sánh và loại *thành phần lặp* của nhóm tác giả bằng cách chia trang web thành 5 phần trên, dưới, trái phải (vùng biên) và giữa (vùng giữa) là không tổng quát và thiếu tính chính xác. Từ đó, chúng em có nhu cầu đề ra phương pháp so sánh độ tương đồng cấu trúc nhằm loại bỏ một cách chính xác hơn các *thành phần lặp*.

* Giải pháp

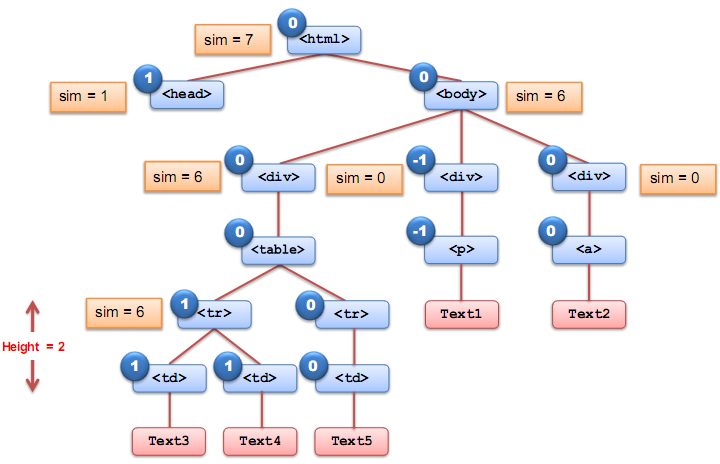
Như đã được phân tích ở mục 6.3.6, cấu trúc thẻ của một trang web hoàn toàn có thể được khôi phục từ xpath của các thẻ chứa văn bản trong trang web. Theo đó, trên cây cấu trúc của một trang web gồm nhiều node tương ứng với các thẻ trong trang web. Một trang web được xem như một node gồm nhiều con. Để đánh dấu độ tương đồng cấu trúc giữa 2 trang web, chúng em sử dụng một số luật sau:

* Hai node được xem như không tương đồng về cấu trúc khi và chỉ khi chúng có số lượng con khác nhau hoặc các con không tương ứng về tên.
* Hai node được xem như tương đồng về cấu trúc khi và chỉ khi chúng có cùng số lượng con và các con hoàn toàn tương ứng giống nhau về tên.
* Hai node được xem như là giống nhau hoàn toàn (*thành phần lặp*) khi và chỉ khi:
  + Tất cả node cha của chúng đều tương đồng về cấu trúc.
  + Có cùng số lượng con và tất cả các con đều giống nhau hoàn toàn về tên lẫn nội dung văn bản.



Hình ‑ So sánh và đánh dấu thành phần lặp giữa 2 cây cấu trúc HTML

Theo đó, các node không tương đồng về cấu trúc được đánh dấu -1, tương đồng về cấu trúc được đánh dấu 0 và giống nhau hoàn toàn được đánh dấu 1. Hình 6‑11 là một ví dụ đánh dấu *thành phần lặp* giữa 2 cây cấu trúc. Trên đó, có nhánh thành phần lặp được gán giá trị 1 vì có 2 thẻ con tương đồng nhau về tên và nội dung *Text*3 và *Text*4. Độ tương đồng của cây cấu trúc được tính dựa trên những chỉ số này sẽ được trình bày ngay tiếp sau đây.



Hình ‑ Tính độ tương đồng cây cấu trúc HTML

Hình 6‑12 mô tả cách tính độ tương đồng cấu trúc của một văn bản HTML. Độ tương đồng của cây cấu trúc HTML được tính dựa trên các nhánh lặp giữa 2 trang web. Theo đó, có 2 yếu tố ảnh hưởng đến độ tương đồng:

* Chiều cao của nhánh lặp: số tầng phân cấp từ node gốc của nhánh lặp đến lá sâu nhất.
* Trọng lượng của nhánh lặp: tổng số node kể cả lá có trong nhánh lặp đó.

Các luật để tính độ tương đồng của 2 nhánh cấu trúc HTML được áp dụng là:

* Nếu node gốc của nhánh được đánh dấu -1 thì độ tương đồng được gán bằng 0.
* Nếu node gốc của nhánh được đánh dấu 0 thì độ tương đồng được gán bằng tổng độ tương đồng các nhánh con.
* Nếu node gốc của nhánh được đánh dấu 1 và cha của node này được đánh dấu 1 thì độ tương đồng được tính bằng trọng lượng của nhánh (tổng số node con của nhánh này).
* Nếu node gốc của nhánh được đánh dấu 1 và cha của node này được đánh dấu 0 thì độ tương đồng được tính bằng tích chiều cao và trọng lượng của nhánh.

### Loại trừ các phần trùng nhau giữa các trang web trong cùng một nhóm.

* Vấn đề

Sau khi phân nhóm các trang web theo độ tương đồng cấu trúc, vấn đề đặt ra là lựa chọn phương thức loại bỏ *thành phần lặp* giữa các trang web trong nhóm sao cho hiệu quả nhất. Nghĩa là tổng độ tương đồng các *thành phần lặp* được loại bỏ là nhiều nhất.

* Giải pháp

Giải pháp của chúng em là chọn ra một trang web làm *hạt giống* (seed) để so sánh độ tương đồng với tất cả các trang web còn lại bằng cách xét một số ngẫu nhiên các trang web làm ứng viên *hạt giống* trên một tập ngẫu nhiên các mẫu thử trong nhóm. Trang web được lựa chọn làm *hạt giống* cần thỏa những tiêu chí sau:

* Trung bình độ tương đồng của ứng viên với các mẫu thử nằm trong nhóm các ứng viên có tổng độ tương đồng lớn nhất. Tiêu chí này đảm bảo tổng số thành phần lặp bị loại bỏ là lớn nhất.
* Phân bố các độ tương đồng giữa ứng viên với các mẫu thử là ổn định nhất dựa vào độ lệch chuẩn (Standard Deviation). Tiêu chí này đảm bảo các thành phần lặp được loại bỏ khỏi các trang web là đồng đều nhất.

### Lưu trữ các trang web vào hệ quản trị cơ sở dữ liệu

* Vấn đề:

Với yêu cầu phân tích và xử lý cấu trúc các trang HTML - được cấu thành bởi các thẻ và phần nội dung văn bản kèm theo thẻ. Với các Website lớn, mỗi trang web có chứa rất nhiều thẻ, trung bình một trang web chứa xấp xỉ hơn 500 thẻ mà số lượng trang web được lưu trữ lên đến hàng chục ngàn. Vì vậy, việc lưu trữ tất cả các thẻ này vào cơ sở dữ liệu thực sự là một gánh nặng.

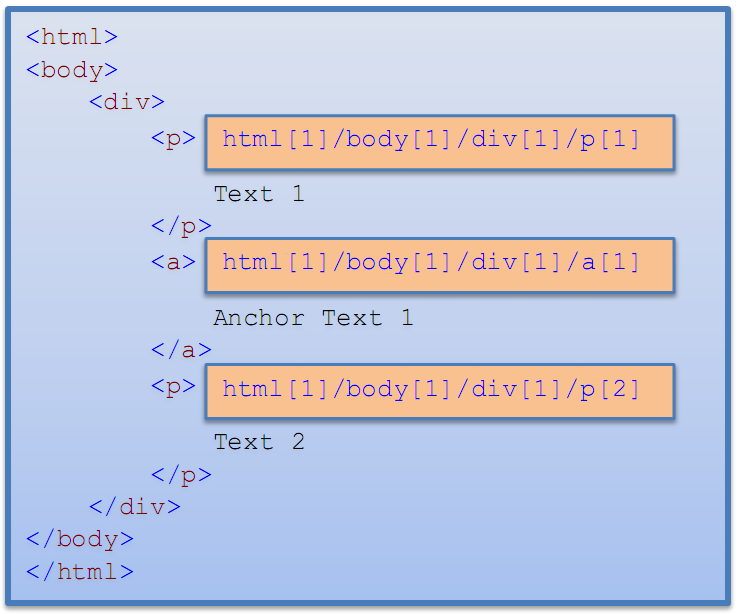
* Giải pháp:

Dựa trên đặc thù yêu cầu của phương pháp, giải pháp cho vấn đề trên phải thỏa mãn 2 yêu cầu thiết yếu sau:

* Có khả năng xây dựng lại cấu trúc thẻ của trang web như ban đầu với số lượng record lưu trữ là ít nhất có thể.
* Phải đảm bảo lưu trữ được tất cả các phần văn bản của trang web.

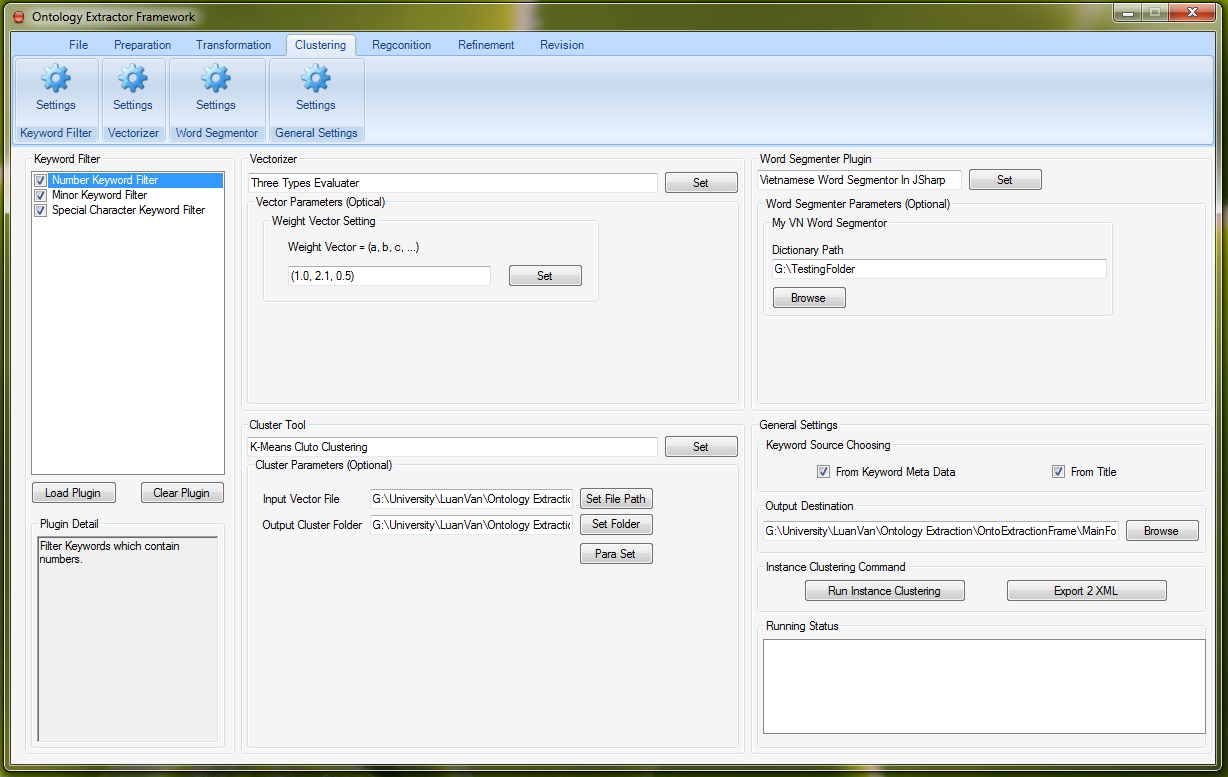
Từ nhận xét các phần văn bản luôn là phần cuối cùng trong một nhánh thẻ của một trang HTML, đặt xpath của một thẻ là đường đi từ thẻ root đến thẻ đó, thẻ root của một văn bản HTML thường là . Giải pháp được chúng em đề ra là chỉ cần lưu trữ xpath của các thẻ có chứa văn bản kèm theo. Như vậy cấu trúc của một trang web có thể hoàn toàn được khôi phục dựa vào xpath của tất cả các thẻ chứa văn bản. Việc lưu trữ tất cả các thẻ chứa văn bản đảm bảo lưu trữ được tất cả các phần văn bản của trang web, tạo tiền đề cho việc thống kê tần số xuất hiện các từ khóa được đầy đủ và trọn vẹn.

Hình 6‑13 cho thấy cách lưu trữ các xpath của một văn bản HTML, trong đó chỉ những thẻ chứa văn bản mới được lưu trữ xpath. Mỗi xpath gồm một dãy các thẻ thể hiện đường đi từ thẻ *root* đến thẻ đang được xét. Mỗi thẻ trong dãy xpath có thêm chỉ số thể hiện thứ tự xuất hiện của mình trong số các thẻ đồng cấp đồng loại. Khi lưu trữ trong cơ sở dữ liệu, các xpath của một văn bản HTML được lưu tuần tự liên tiếp nhau, điều này đảm bảo cho quá trình khôi phục cấu trúc thẻ của văn bản dựa vào tập các xpath này.



Hình ‑ Xpath

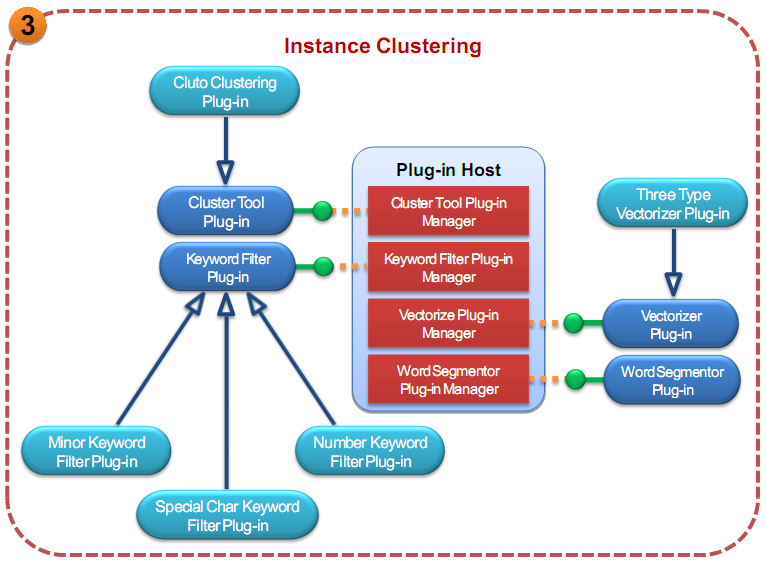
## Phân hệ Gom cụm (Instance Clustering)



Hình ‑ Giao diện phân hệ Gom cụm

### Kiến trúc phân hệ

Các trang web sau khi được lọc bỏ các thành phần không cần thiết ở bước *Phân hệ* Biến đổi được biểu diễn bằng các *vector đặc trưng* phản ánh nội dung bên trong của các trang web. Nhiệm vụ chính của phân hệ Gom cụm chính là tiến hành gom cụm các trang web dựa vào độ tương đồng về nội dung thông qua *vector đặc trưng*. Kiến trúc của phân hệ sử dụng cơ chế plug-in được trình bày ở Hình 6‑15.



Hình ‑ Kiến trúc Phân hệ Gom cụm

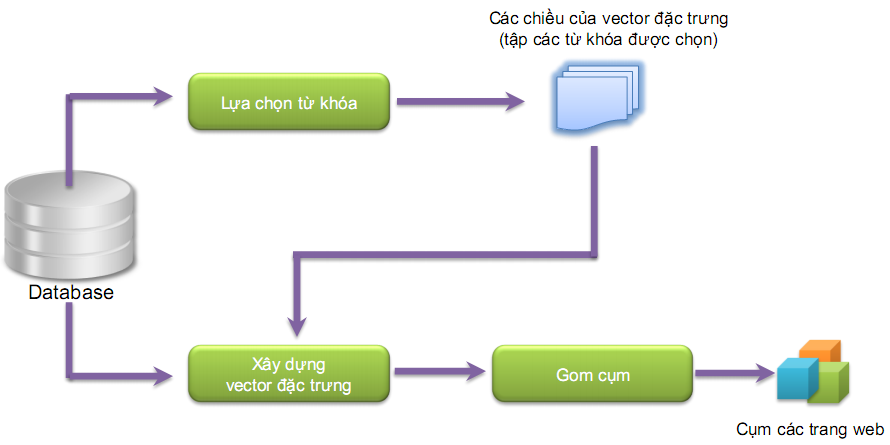
Hình 6‑15 mô tả kiến trúc của phân hệ Chuẩn bị. Phân hệ gồm một host quản lý các plug-in và có 4 loại plug-in. Trong đó chúng em đã xây dựng các thể hiện cho mỗi loại plugin:

* Cluster Tool Plug-in: hỗ trợ công cụ gom cụm các vector đặc trưng. (1 thể hiện)
* Keyword Filter Plug-in: bộ lọc các từ khóa. (3 thể hiện)
* Vectorizer Plug-in: xây dựng vector đặc trưng cho các trang web. (1 thể hiện)
* Word Segmentor Plug-in: công cụ tách từ tiếng Việt. (2 thể hiện: tái sử dụng của phân hệ Biến đổi)

Bảng 6‑4 trình bày các lớp đối tượng chính cấu thành phân hệ Chuẩn bị.

|  |  |  |
| --- | --- | --- |
| STT | Tên Lớp | Mô tả |
| 1 | IinstanceClusteringHost | Interface mô tả host quản lý các plug-in trong phân hệ Gom Cụm |
| 2 | InstanceClusteringHost | Thể hiện host của phân hệ Gom cụm |
| 3 | Iclustering | Interface mô tả plugin gom cụm các vector đặc trưng |
| 4 | ClutoClusteringPlugin | Plugin gom cụm sử dụng thư viện Cluto |
| 5 | Ivectorizer | Interface mô tả plugin xây dựng vector đặc trưng |
| 6 | ThreeTypeVectorizerPlugin | Plugin xây dựng vector đặc trưng trên cơ sở phân loại văn bản thành 3 loại |
| 7 | IkeywordFilter | Interface mô tả plugin bộ lọc từ khóa |
| 8 | NumberKeywordFilterPlugin | Bộ lọc các các từ khóa có chứa chữ số |
| 9 | SpecialCharKeywordFilterPlugin | Bộ lọc các từ khóa có chứa ký tự đặc biệt |
| 10 | MinorKeywordFilterPlugin | Bộ lọc các từ khóa có tần số xuất hiện thấp |

Bảng 6‑4 Các lớp chính cấu thành phân hệ Gom cụm



Hình ‑ Mô hình xử lý tổng quát Phân hệ Gom cụm

Như đã được phân tích ở mục **Error! Reference source not found.**, các tác giả không nhắc đến cách xây dựng các vector đặc trưng *cùng chiều* và *tương ứng từ khoá*, vì thế chúng em đề nghị giải pháp cho vấn đề này bằng cách chọn ra một tập các từ khoá ứng viên trong số tất cả các từ khoá đã được gán nhãn cho các trang web. Tập các từ khoá ứng viên này được dùng làm các chiều chuẩn chung của *vector đặc trưng* cho tất cả các trang web. Chi tiết phương pháp chọn từ khoá ứng viên được mô tả ở mục 6.4.2.

Hình 6‑16 cho thấy, trước khi thực hiện việc xây dựng vector, bộ lọc từ khoá làm nhiệm vụ lựa chọn ra tập các từ khoá ứng viên làm số chiều chuẩn chung cho tất cả các vector đặc trưng. Cuối cùng, các trang web được gom cụm thông qua việc gom cụm các vector đại diện của chúng.

### Lựa chọn các từ khoá

* Vấn đề

Giải pháp được đề ra nhằm đảm bảo yêu cầu các vector đặc trưng phải *cùng chiều* và *tương ứng từ khoá* là chọn ra một tập các từ khoá ứng viên. Vì tổng số lượng từ khoá được lưu trữ là rất lớn, nên chỉ những từ khoá nào có triển vọng nhất mới được chọn.

* Giải pháp

Giải pháp của chúng em là xây dựng bộ lọc các từ khoá, với các tiêu chí động nhằm hỗ trợ người chuyên gia ontology lựa chọn được tập các ứng viên ưng ý nhất. Một số tiêu chí đã được chúng em xây dựng được trình bày ở bảng:

* Chỉ lấy từ khoá là chữ.
* Chỉ lấy từ khoá không có ký tự đặc biệt.
* Chỉ lấy từ khoá có tần số xuất hiện cao.

|  |  |  |
| --- | --- | --- |
| STT | Bộ lọc từ khóa | Mô tả |
| 1 | NumberKeywordFilterPlugin | Bộ lọc các các từ khóa có chứa chữ số |
| 2 | SpecialCharKeywordFilterPlugin | Bộ lọc các từ khóa có chứa ký tự đặc biệt |
| 3 | MinorKeywordFilterPlugin | Bộ lọc các từ khóa có tần số xuất hiện thấp |

Bảng 6‑5 Các bộ lọc từ khóa

Trong đó, bộ lọc các từ khoá có tần số xuất hiện cao đã được chúng em xây dựng và triển khai bằng cách thống kê số trang web chứa từ khoá này. Chỉ những từ khoá có số lần xuất hiện trong các trang web lớn hơn một ngưỡng ε mới được chọn. Ngưỡng này được điều chỉnh bởi người chuyên gia ontology như đã được đề cập ở mục 0.

Phương pháp xây dựng vector đặc trưng của các trang web được hiện thực hoá và cải tiến dựa trên phương pháp của nhóm tác giả như đã được đề cập ở mục **Error! Reference source not found.**. Trong đó, giá trị các chiều của *vector trọng số* cần được điều chỉnh tuỳ vào người chuyên gia ontology như sẽ được đề cập ở mục 0.

### Xây dựng vector đặc trưng

* Vấn đề

Phương pháp được sử dụng trong luận văn này dựa trên tần số xuất hiện của các từ khóa (chọn từ thẻ , ) trong phần văn bản nội dung của trang web. Do đó, nhu cầu tách từ phần văn bản chính nhằm thống kê tần số là một trong những vấn đề cần được giải quyết trong bước *gom cụm* này.

* Giải pháp

Một trong những phương pháp tách từ tiếng Việt được xem là hiệu quả và nhanh nhất hiện này là phương pháp so khớp từ dài nhất (Longest Matching). Theo đó, để tách một phần văn bản, bộ từ điển có sẵn được sử dụng và đánh dấu các từ trong phần văn bản theo các từ có trong từ điển với tiêu chí ưu tiên các từ dài nhất trước. Lần lượt đánh dấu tất cả các từ có thể có, phần còn lại được phân biệt bởi các từ đã được đánh dấu được xem như một từ. Từ đây, có hai hướng tiếp cận:

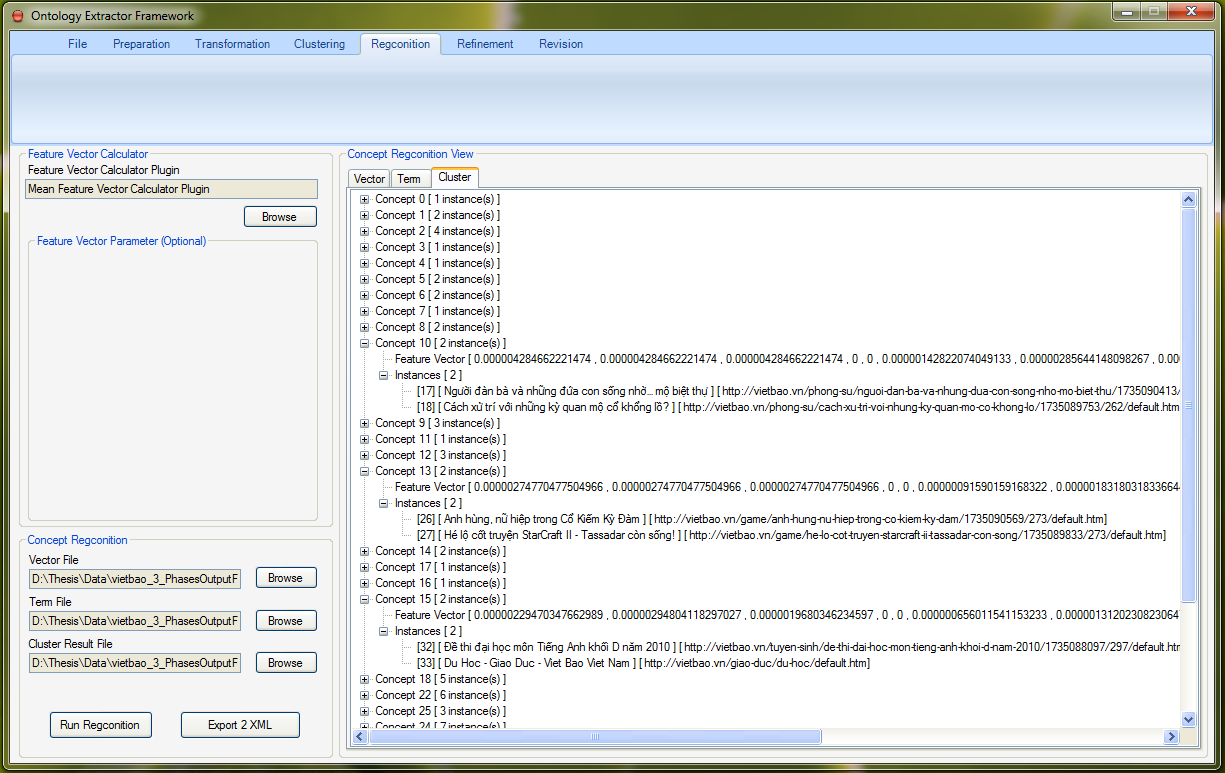
* Sử dụng bộ từ điển chung, được xây dựng sẵn một cách phân biệt với hệ thống.
* Sử dụng tập các từ khóa ứng viên được chọn của Website làm từ điển.

Trong thực tế hiện thực hoá, chúng em nhận thấy việc sử dụng tập các từ khoá ứng viên được chọn của Website làm từ điển mang lại kết quả tốt hơn, vì các từ tách được chính xác với các từ khoá, do đó kết quả tính toán tần số sẽ tốt hơn.

### Gom cụm các trang web dựa trên độ tương đồng giữa các vector đặc trưng

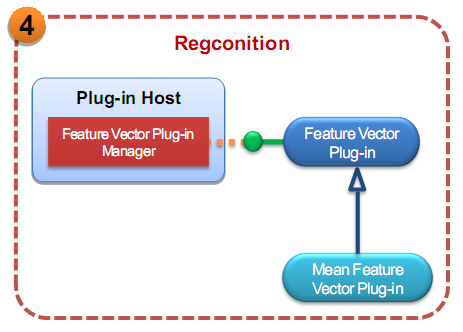
Đặc thù của việc gom cụm các vector đặc trưng là số chiều rất lớn. Công cụ CLUTO là một công cụ gom cụm miễn phí để thực hiện gom cụm các vector (<http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview>). Một trong các ưu điểm của CLUTO mà chúng em quan tâm là CLUTO có thể thực hiện gom cụm trên các phần tử có số chiều rất lớn. Điều này giải quyết được vấn đề gom cụm các vector đặc trưng có số chiều lớn của phương pháp. Vì thế, chúng em lựa chọn CLUTO như là giải pháp cho vấn đề gom cụm.

## Phân hệ Nhận diện (Recognition)



Hình ‑ Giao diện phân hệ Nhận diện

### Kiến trúc phân hệ



Hình ‑ Kiến trúc Phân hệ Nhận diện

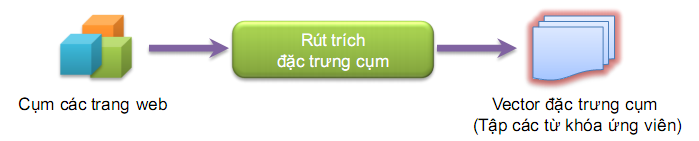
Hình 6‑18 mô tả kiến trúc của phân hệ Nhận diện. Phân hệ gồm một host quản lý các plug-in và có 1 loại plug-in. Trong đó chúng em đã xây dựng các thể hiện cho loại plugin này:

* Feature Vector Plug-in: rút trích đặc trưng cụm. (1 thể hiện)

Bảng 6‑3 Các lớp chính cấu thành phân hệ Biến đổi trình bày các lớp đối tượng chính cấu thành phân hệ Biến đổi.

|  |  |  |
| --- | --- | --- |
| STT | Tên Lớp | Mô tả |
| 1 | IregconitionHost | Interface mô tả host quản lý các plug-in trong phân hệ Nhận diện |
| 2 | RegconitionHost | Thể hiện host của phân hệ Nhận diện |
| 3 | IfeatureVectorCalculator | Interface mô tả plugin rút trích đặc trưng cụm |
| 4 | MeanFeatureVectorPlugin | Plugin rút trích đặc trưng cụm sử dụng trung bình cộng |

Bảng 6‑6 Các lớp chính cấu thành phân hệ Nhận diện



Hình ‑ Mô hình xử lý tổng quát Phân hệ Nhận diện

Hình 6‑22 mô tả 2 bước của quá trình *nhận diện*. Trong đó, kết quả quan trọng nhất của bước *nhận diện* chính là *vector đặc trưng cụm*. Mỗi cụm tương ứng với một *khái niệm*. Từ *vector đặc trưng cụm*, các từ khoá ứng viên của *khái niệm* được rút ra. Trong đó, các chiều của *vector đặc trưng cụm* có giá trị lớn hơn giá trị ε do chuyên gia ontology điều chỉnh (mục 0) sẽ được rút làm *từ khoá ứng viên* của *khái niệm*.

### Nhận diện đặc trưng cụm

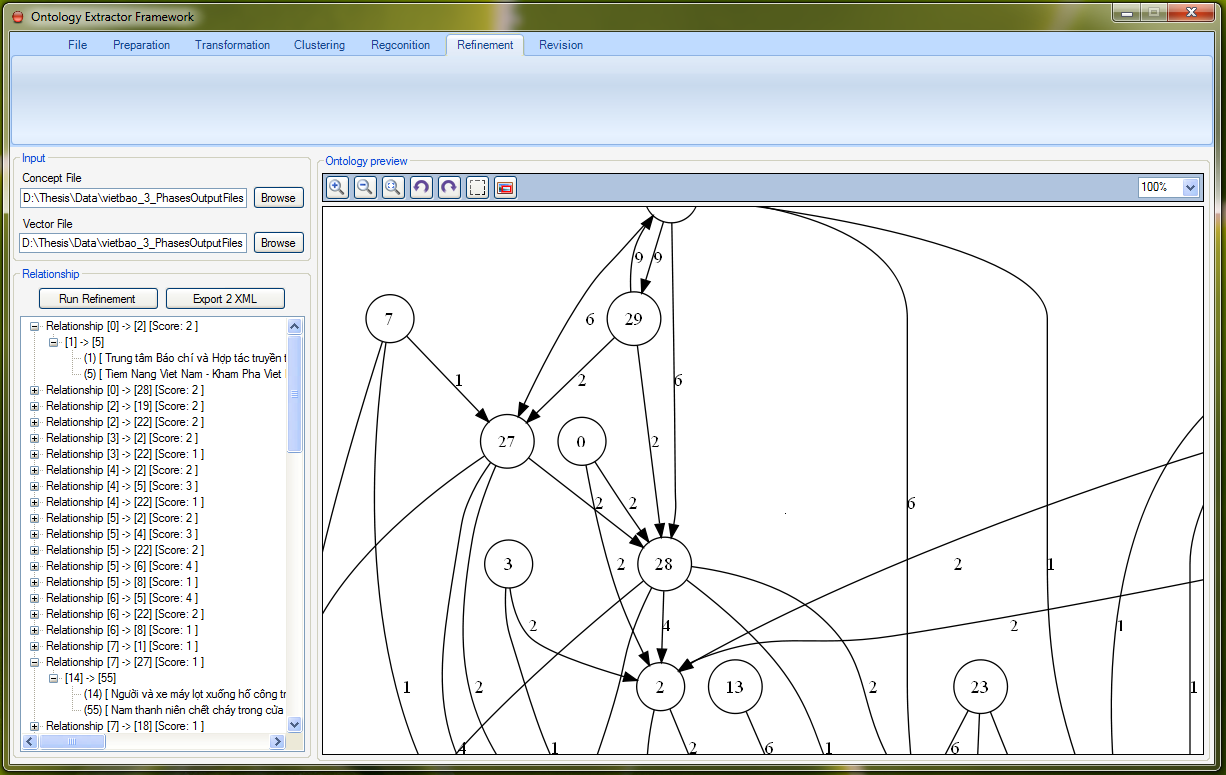
* Vấn đề

Mỗi cụm gồm nhiều trang web tương đồng về nội dung. Mỗi trang web được đại diện bằng một vector đặc trưng, vì vậy *đặc trưng cụm* chính là một vector *cùng chiều* và *tương ứng từ khóa* với các vector thành phần.

* Giải pháp

Mỗi phần tử được biểu diễn bằng một vector n chiều. Giải pháp được chúng em đề nghị và triển khai trong thực tế là rút trích các vector đại diện cho các cụm. Trong đó, giá trị mỗi chiều của vector đại diện là trung bình cộng (mean) các giá trị tại chiều tương ứng của các vector phần tử.

## Phân hệ Liên kết (Refinement)



Hình ‑ Giao diện phân hệ Liên kết

### Kiến trúc phân hệ



Hình ‑ User Control phân hệ Liên kết

Kiến trúc phân hệ chỉ gồm một User Control là RefineUC đóng vai trò hỗ trợ giao diện trong việc xây dựng đồ thị các *mối quan hệ*.



Hình ‑ Mô hình xử lý tổng quát Phân hệ Liên kết

Ở bước *Phân hệ* Nhận diện, các *khái niệm* đã xác định được các từ khoá ứng viên tương ứng. Hình 6‑22 mô tả 2 bước thực hiện quá trình *Liên kết*. Đầu tiên, các *mối quan hệ* giữa các *khái niệm* được xác định và gán trọng số. Kết quả của bước này thu được một đồ thị với các node là các *khái niệm* được đại diện bằng tập các từ khoá ứng viên. Các cạnh của đồ thị có hướng và được gán trọng số. Sau đó, đồ thị được tinh chỉnh nhằm thu được kết quả chính xác và thuận lợi hơn cho người chuyên gia ontology ở bước Phân hệ Tinh chỉnh tiếp theo.

### Xây dựng đồ thị mối quan hệ giữa các khái niệm

* Vấn đề

Mỗi *khái niệm* được đại diện bằng một cụm mà các phần tử trong đó là các trang web. Vấn đề đặt ra là cần xác định các *mối quan hệ* giữa các *khái niệm*. Hai trang web được xem là có quan hệ khi có đường dẫn từ trang này qua trang kia. Như vậy, *mối quan hệ* giữa hai *khái niệm* được xác định bằng đường dẫn giữa các trang web thuộc về 2 *khái niệm* đó. Vấn đề đặt ra chính là các *khái niệm* và *mối quan hệ* này chỉ đang ở dạng thô sơ, cần được biểu diễn sao cho có thể hỗ trợ tối đa nhất cho việc quan sát, xử lý và tinh chỉnh sau này của chuyên gia ontology.

* Giải pháp

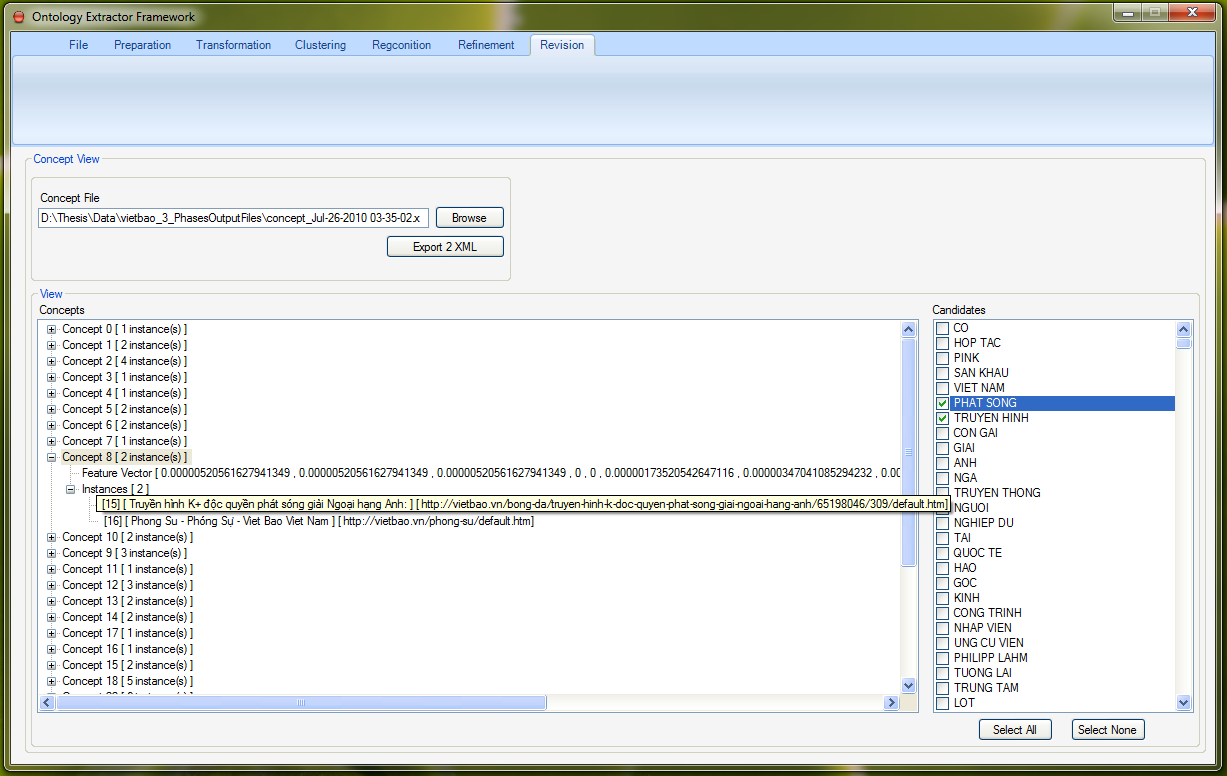
Giải pháp được đặt ra là biểu diễn các *khái niệm* và *mối quan hệ* dưới dạng đồ thị. Đồ thị mối quan hệ do chúng em đề nghị xây dựng mang những đặc điểm:

* Mỗi đỉnh trên đồ thị đại diện cho một *khái niệm* (cụm các trang web).
* Đồ thị có hướng và các cạnh được đánh trọng số.
* Hai đỉnh được nối với nhau khi tồn tại một đường dẫn từ trang web thuộc cụm này dẫn đến một trang web thuộc cụm kia. Trọng số của cạnh chính là số lượng đường dẫn nối hai cụm.

Với giải pháp này, cách hiện thực hóa của chúng em mang tính cải tiến so với phương pháp của các tác giả như đã được nhắc đến ở mục **Error! Reference source not found.**. Trong đó :

* Do nhận nhận xét các *mối quan hệ* có tính *duy nhất* và *đối xứng* của các tác giả, các *mối quan hệ* (đại diện bằng các cạnh trên đồ thị) sẽ không được gán trọng số và vô hướng. Ngược lại, chúng em lưu trữ trọng số và hướng của các mối quan hệ sẽ giúp cung cấp thêm thông tin cho các chuyên gia ontology để thuận lợi cho việc tinh chỉnh sau này.
* Nhận xét các *mối quan hệ* có tính *bắc cầu* khiến các cạnh nối giữa 2 đỉnh sẽ bị xóa đi nếu tồn tại một đường đi vô hướng khác dài hơn từ đỉnh này đến đỉnh kia. Chúng em vẫn lưu giữ lại những cạnh nối trực tiếp này, vì như vậy, quan hệ giữa 2 đỉnh có cạnh nối trực tiếp vẫn có ý nghĩa hơn là mối quan hệ được suy ra từ tính chất bắc cầu.

## Phân hệ Tinh chỉnh (Revision)



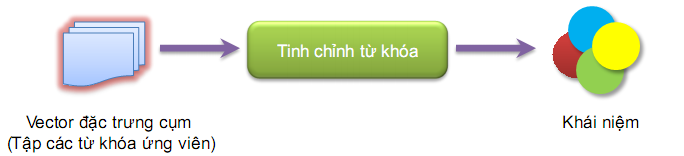
Hình ‑ Giao diện phân hệ Tinh chỉnh

### Kiến trúc phân hệ



Hình ‑ User Control phân hệ Tinh chỉnh

Kiến trúc phân hệ chỉ gồm một User Control là RevisionUC đóng vai trò hỗ trợ giao diện trong việc lựa chọn các từ khóa đại diện cho *khái niệm*.



Hình ‑ Mô hình xử lý tổng quát Phân hệ Tinh chỉnh

Người chuyên gia ontology đóng vai trò quan trọng trong việc vận hành toàn bộ hệ thống, điều chỉnh các thông số và tinh chỉnh các *khái niệm*, *mối quan hệ* cuối cùng nhằm đảm bảo kết quả ontology của hệ thống sao cho phù hợp nhất với mục đích sử dụng trong tương lai của ontology. Đối với hệ thống được triển khai, chuyên gia ontology là người lựa chọn các plug-in của hệ thống và điều chỉnh các tham số cần thiết cho các plug-in cần tham số. Cụ thể như sau:

* Điều chỉnh tham số ngưỡng tinh chỉnh số lượng trang web trong nhóm (mục 6.3.3)
* Điều chỉnh ngưỡng để lựa chọn từ khoá ứng viên triển vọng cho *vector đặc trưng* (mục 6.4.2).
* Điều chỉnh các tham số giá trị của *vector trọng số* trong việc xây dựng *vector đặc trưng* của trang web (mục 6.4.3).

# Kết luận

* Nội dung của Chương 7 trình bày các kết quả đạt được và hướng phát triển của đề tài.

## Các kết quả đạt được

Chúng em đã tiến hành tìm hiểu và nghiên cứu về ontology và các vấn đề liên quan đến ontology. Đồng thời, chúng em đã tìm hiểu các phương pháp khác nhau đê xây dựng ontology, và chú trọng vào các phương pháp rút trích ontology với nguồn dữ liệu lấy từ web.

Qua quá trình nghiên cứu và tìm hiểu đó, chúng em tiến hành phân tích cụ thể quy trình rút trích ontology từ web [4] do nhóm tác giả Du C. Timon, Li Feng, và King Irwin đề xuất. Những phân tích cụ thể về quy trình này cho thấy mức độ tổng quát cao của quy trình giúp cho nó có thể được áp dụng vào nhiều ngữ cảnh khác nhau và có độ tùy biến cao tùy thuộc vào người sử dụng.

Do đó, chúng em đã tiến hành xây dựng Ontology Extractor Framework để hiện thực hóa quy trình này. Trong quá trình xây dựng framework này, đề tài đã nêu ra và phân tích các vấn đề phát sinh khác nhau cũng như đưa ra hướng giải quyết cho các vấn đề này. Đề tài cũng đã xây dựng kiến trúc của hệ thống này phục vụ việc tùy biến theo nhu cầu cụ thể của người dùng.

Ngoài framework được xây dựng, chúng em còn tiến hành hiện thực hóa các plug-in cụ thể để sử dụng trên framework này. Trong quá trình hiện thực hóa này, chúng em đã phân tích các vấn đề nảy sinh và đề ra hướng giải quyết cho những vấn đề này mà nhóm tác giả của quy trình này không đề cập đến.

## Hướng phát triển của đề tài

Những dự định trong tương lai để phát triển framework này là:

* Hoàn thiện các chức năng hiện có của framework.
* Tăng cường tính tiện dụng cho framework.
* Xây dựng các bộ plug-in khác nhau để hoạt động trên framework này.

Danh mục tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | InfoWatch, "Global Data Leakages & Insider Threats Report - 2012," InfoWatch Research Center, 2012. |
| [2] | InfoWatch, "Global Data Leakage Report - 2013," InfoWatch Analytical Center, 2014. |
| [3] | InfoWatch, "Infowatch's Global Data Leakage Report for First Half of 2014," 2014. |
| [4] | D. C. Timon, L. Feng and K. Irwin, "Managing knowledge on the Web - Extracting ontology from HTML Web," *Decision Support Systems,* vol. 47, no. 4, pp. 319-331, November 2009. |
| [5] | McAfee, "McAfee for Small Business," [Online]. Available: http://www.shopmcafee.com/store/mfesmb/en\_US/DisplayHomePage?CID=MFE-MHP102. |
| [6] | McAfee, "McAfee for Business," [Online]. Available: http://www.mcafee.com/us/business-home.aspx?CID=MFEen-usMHP002. |
| [7] | TrendMicro, "Worry-Free Business Security," [Online]. Available: http://www.trendmicro.com/us/small-business/product-security/index.html. |
| [8] | TrendMicro, "Enterprise Security Software," [Online]. Available: Worry-Free Business Security. |
| [9] | TrendMicro, "Integrated Data Loss Prevention," [Online]. Available: http://www.trendmicro.com/us/enterprise/data-protection/data-loss-prevention/. |
| [10] | Symantec, "Products & Solutions," [Online]. Available: http://www.symantec.com/products-solutions/. |
| [11] | Symantec, "Data Loss Prevention," [Online]. Available: http://www.symantec.com/data-leak-prevention/. |
| [12] | EMC, "RSA DATA LOSS PREVENTION," [Online]. Available: http://www.emc.com/security/rsa-data-loss-prevention.htm. |
| [13] | WebSense, "TRITON APX Core Products," [Online]. Available: http://www.websense.com/content/triton-apx.aspx?intcmp=nav-mm-products-core-apx. |
| [14] | I. Ivanov, "API hooking revealed," 2 12 2002. [Online]. Available: http://www.codeproject.com/Articles/2082/API-hooking-revealed. |
| [15] | Wikipedia, "http://en.wikipedia.org," [Online]. Available: http://en.wikipedia.org/wiki/Portable\_Executable. |
| [16] | J. Xing and T. Ah-Hwee, "Mining Ontological Knowledge from Domain-Specific Text Documents," in *ICDM Proceedings of the Fifth IEEE International Conference on Data Mining*, Washington, DC, 2005. |
| [17] | S. York, B. Stephan, H. Peter, H. Jens and O. Daniel, "The SWRC Ontology - Semantic Web for Research Communities," in *Progress in Artificial Intelligence, 12th Portuguese Conference on Artificial Intelligence*, 2005. |
| [18] | A. B. Williams and C. Tsatsoulis, "An Instance-based Approach for Identifying Candidate Ontology Relations within a Multi-Agent System," in *Proceedings of the First Workshop on Ontology Learning OL'2000*, Berlin, 2000. |