

Phần A – Tài liệu kỹ thuật VietSpider.

I - Viễn cảnh về khai thác thông tin.

1. Điểm lược một vài mô hình khai thác và tổng hợp nội dung

II - Giới thiệu về phần mềm.

1. Yêu cầu bài toán về khai thác và tổng hợp nội dung.
2. Giới thiệu về phần mềm.
3. Một số thông tin cơ bản.
4. Một số yêu cầu cơ bản khi chạy phần.

III - Sơ lược về cơ sở kỹ thuật trong chương trình.

1. Khảo sát định dạng phát hành nội dung.
2. Sơ lược về mô hình bóc tách dữ liệu.
3. Kỹ thuật về tổng hợp nội dung.

IV - Những thành phần cơ bản của chương trình.

1. Bộ HTMLParser và công cụ HTML Explorer.
2. Công cụ cấu hình một kênh khai thác thông tin.
3. Công cụ bóc tách và tổng hợp nội dung.
4. Giải pháp phát hành nhanh nội dung.

V - Mô tả cơ sở dữ liệu cho chương trình.

1. Bảng Domain.
2. Bảng Meta.
3. Bảng Content.
4. Bảng Relation.
5. Bảng Image.
6. Bảng Filter.

VI - Ý nghĩa thư mục dữ liệu.

1. Làm sạch dữ liệu sau bóc tách.
2. Cấu hình một số thông số cho chương trình.

VII - Tài liệu và địa chỉ tham khảo.

I. Viễn cảnh về khai thác thông tin.

Sự phát triển của mạnh mẽ của Internet kéo theo hàng loạt những mô hình truyền thông mạng như báo điện tử, blog, forum, trang thông tin cá nhân, tổ chức, cơ quan, công ty,...Tiếp cận nguồn thông tin phong phú đó làm nảy sinh một nhu cầu: khai thác và tổng hợp hiệu quả các nội dung từ Internet.

1. Điểm lược một vài mô hình khai thác và tổng hợp nội dung.

Thông tin cũng là một tài nguyên cần khai thác và Internet giống như một mỏ thông tin khổng lồ được cập nhật từng giờ từng phút. Khai thác thông tin là một cụm từ xuất hiện trước đó rất lâu so với thời điểm ra đời của Internet. Hiện nay, sự khai thác thông tin từ Internet đã là một nhu cầu của mỗi cá nhân. Không quá xa vời, những phóng viên báo chí hằng ngày vẫn tìm kiếm tư liệu, tham khảo các bài viết hoặc thậm chí đăng lại nội dung từ một nguồn cụ thể như website báo điện tử, blog, diễn đàn... Bằng cách này hay cách khác, họ hằng ngày vẫn đang khai thác thông tin cho công việc và nhu cầu hiểu biết của họ. Do đó, một công cụ trợ giúp việc cập nhật, khai thác và quản lý thông tin hiệu quả là cần thiết.

Có nhiều hình thái về khai thác và tổng hợp nội dung đã được nghiên cứu và phát triển. Chúng ta có một loạt khái niệm như Robot, Search, Web Crawler, Data Wrapper, Web Spider, Web Clipping, Semantic Web,... để mô tả về những hình thái khai thác nội dung thông tin trên Internet. Xin lấy mô hình tìm kiếm là một ví dụ: Nội dung sau khi khai thác có thể được lưu trữ trong các hệ thống database và phát hành lại tới người dùng trực tiếp thông qua hệ thống tích hợp, tìm kiếm, lọc, chia sẻ đặt tải,...hay sử dụng cho một mục đích chuyên biệt đó. Google là minh chứng cụ thể cho giải pháp đó, các Website tồn tại trên Internet sẽ được Google Crawler ghé thăm và thu thập lại toàn bộ, sau đó nội dung được lưu trữ trong cơ sở dữ liệu, được đánh chỉ mục,... và được tìm kiếm mỗi khi có yêu cầu từ phía người dùng. Một sản phẩm khác là GoogleNews lại có nhiệm vụ tổng hợp tất cả các tin tức diễn ra hàng ngày trên Internet. Ở Việt nam, ta có thể tìm kiếm những mô hình tương tự như Baomoi.com hay Thegioitin.com. Ngoài ra, chúng ta còn có những chuẩn về chia sẻ đặc tả nội dung như RSS, RDF, Atom,... chúng kết nối thông tin giữa những website và cũng cho phép người dùng tổng hợp các đặc tả bằng những công cụ chuyên biệt như RSS Reader. Như vậy, thực tế cho ta thấy, đã có rất nhiều những mô hình khai thác và tổng hợp nội dung.

II. Giới thiệu về phần mềm.

1. Yêu cầu bài toán về khai thác và tổng hợp nội dung.

Sự phát triển của thông tin tiếng Việt trên mạng Internet và nhu cầu khai thác tổng hợp những nội dung đó. Như đã nói ở phần I, không có gì mới lạ về mặt ý tưởng và cũng đã có những phần mềm ra đời như một thử nghiệm của sự tìm tòi hay ý tưởng kinh doanh. Đã có những thành công nhất định, nhưng thị trường cũng không phải là sự độc quyền của chỉ một sản phẩm phần mềm. Sẽ nảy sinh nhiều phần mềm khác nữa với những chức năng tương tự. Ý tưởng ban đầu cho ứng dụng khai thác và tổng hợp nội dung. Giải pháp đưa ra chủ yếu tập trung xây dựng phần back-end (chương trình phụ trợ) hoặc dành cho người dùng đầu cuối, là một ứng dụng dạng Desktop. Giải pháp có nhiệm vụ khai thác và tổng hợp trực tiếp rồi lưu trữ vào cơ sở dữ liệu. Những thành phẩm sẽ là đầu vào cho những hệ thống được xây dựng với mục đích khác nhau nhưng cùng có chung yêu cầu là cần nội dung phát hành trên Internet.

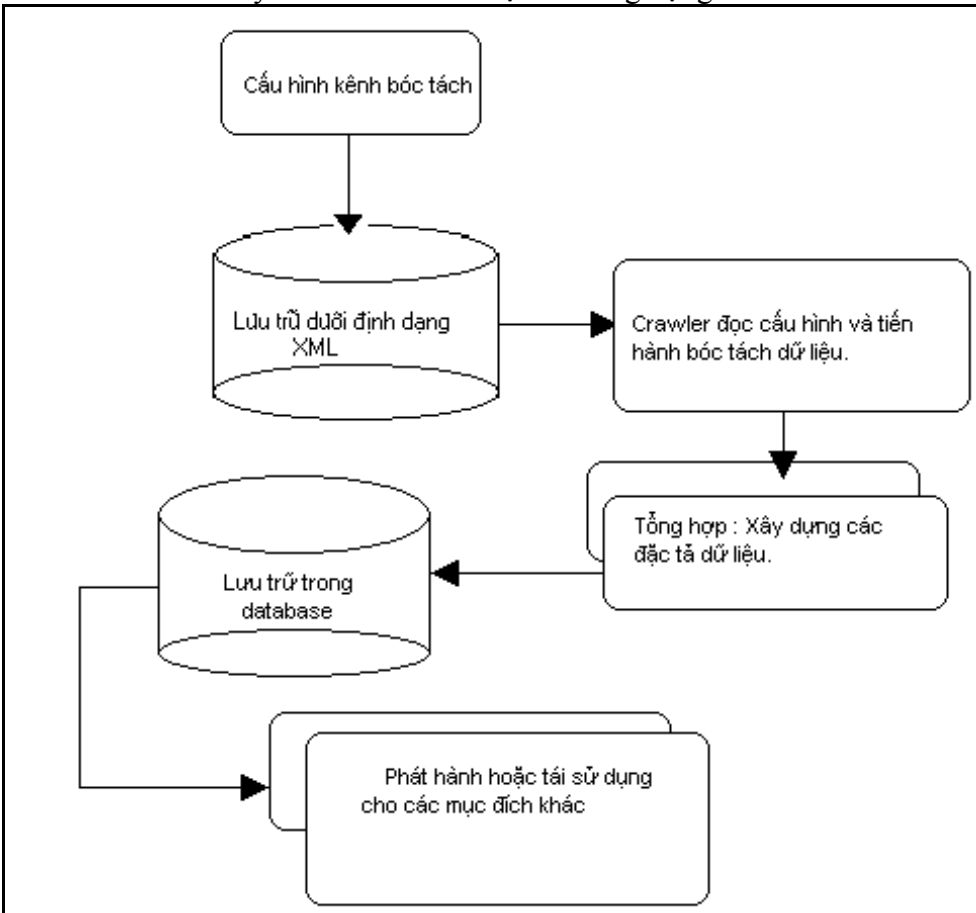
Phần A – Tài liệu kỹ thuật VietSpider

Công cụ xây dựng cũng phải đảm bảo chức năng cấu hình một kênh khai thác mới với sự đơn giản, thuận tiện và nhanh chóng. Hệ thống tổng hợp có khả năng kiểm soát được những nội dung liên quan cùng những nội dung được đăng tải lại giúp cho sự theo dõi có hệ thống các sự kiện xảy ra hàng ngày.

2. Giới thiệu về phần mềm.

Cũng giống như Google News, hệ thống khai thác và tổng hợp nội dung có nhiệm vụ khai thác, tổng hợp, lưu trữ rồi phát hành lại tới người dùng. Crawler nhận cấu hình đầu vào của một website (tin tức, blog, ...) tiến hành bóc tách, tổng hợp chủ đề liên quan, lưu trữ trong database và phát hành lại tới người đầu cuối.

Giải pháp đề xuất không dựa trên mô hình trích xuất dữ liệu giống như các chuẩn RSS, ATOM... hay các mô hình khác dựa trên nền XML được dùng với mục đích chia sẻ đặc tả dữ liệu của nội dung (còn gọi là meta data - cung cấp các thông tin cơ bản bao gồm : tên tin bài, ngày phát hành, sơ lược nội dung, người viết,...). Nội dung được bóc tách toàn vẹn, sạch sẽ và được tổng hợp từ nhiều nguồn khác nhau giúp người đọc có thể theo dõi, kiểm soát, tìm kiếm, biên soạn, lưu trữ một cách hiệu quả. VietSpider là một phần mềm bóc tách đúng nghĩa, chúng truy xuất trực tiếp vào nội dung toàn diện rồi tiến hành bóc tách. Sau đó những đặc tả dữ liệu (meta data) được xây dựng tự động trên nền nội dung đã bóc tách. Sau quy trình khai thác, nội dung sẽ trở thành độc lập với website nguồn, được lưu trữ và tái sử dụng cho những mục đích khác nhau. Dưới đây là mô hình làm việc của ứng dụng.



3. Một số thông tin cơ bản.

VietSpider được phát triển trên nền tảng Java. Sản phẩm nguồn mở sử dụng : Apache HTTPClient-3.x.và các thư viện đi kèm, SWT + JFace 3.x, Apache Lucene 2.x, JChardet Mozilla detect encoding và một số module nhỏ khác... Hỗ trợ các cơ sở dữ liệu thông dụng bao gồm MS SQL Server, MySQL, Oracle, Postgre, Apache Derby. Ứng dụng máy chủ được phát triển và tích trực tiếp trên VietSpider.

4. Một số yêu cầu cơ bản khi chạy phẩm.

Yêu cầu phần cứng:

- Chip Pentium III hoặc cao hơn
- Ổ cứng trống 500 Mb trở lên.
- Ram 256 Mb trở lên .
- Môi trường mạng .

Yêu cầu phần mềm:

- Hệ quản trị cơ sở dữ liệu (mặc định chương trình đã cài sẵn Apache Derby ở dạng tích hợp).
- IE 5.0 trở lên.
- JRE 1.6 trở lên.

III.Sơ lược về cơ sở kỹ thuật trong chương trình.

1. Khảo sát định dạng phát hành nội dung.

Internet ra đời với mục đích chia sẻ thông tin và *World Wide Web* (viết tắt là *Web*) bổ sung cách thức đưa nội dung lên Internet. Tài liệu trên World Wide Web là những văn bản được lưu trữ trong các máy tính kết nối với Internet. Để xem các tài liệu này, người dùng dùng một trình duyệt Web (*Web Browser*) mở và hiển thị chúng. HTML (viết tắt *HyperText Markup Language*) tạm dịch là “Ngôn ngữ đánh dấu siêu văn bản” là một định dạng thông dụng cho tài liệu Web định nghĩa cách thức trình bày, hiển thị nội dung như thế nào ở phía trình duyệt. HTML hiện tại đã trở thành một chuẩn [Internet](#) do tổ chức [World Wide Web Consortium](#) (W3C) duy trì.

Mỗi yêu cầu được gửi từ trình duyệt tới máy chủ sẽ có thể trả về một tài liệu được định dạng bằng HTML, chúng là tập hợp dữ liệu dùng các thẻ được định nghĩa trước đó để quy ước nội dung sẽ được bày bố, hiển thị như thế nào ở phía máy khách. Khi tài liệu này được trình duyệt nhận lại từ server, chúng sẽ chuyển sang một mô hình dữ liệu để sử dụng hơn gọi HTML DOM. DOM là viết tắt của *Document Object Model* tạm dịch là “Mô hình đối tượng tài liệu” có giao diện lập trình ứng dụng (*API*). Thông thường DOM có cấu trúc dạng cây, rất dễ dàng để truy xuất các thành phần trong cây dữ liệu đó, DOM có thể dùng để phân tích HTML, XML hay các định dạng tài liệu khác. Sau khi chuyển đổi tài liệu HTML sang DOM, trình duyệt dùng nó để hiển thị giao diện đồ họa tới người dùng.

2. Sơ lược về mô hình bóc tách dữ liệu.

Tài liệu HTML sau khi được chuyển đổi sang cây DOM (Tree DOM) sẽ dễ dàng truy xuất những thành phần nội dung cần quan tâm thông qua việc truy xuất các nhánh của cây. Nhiều mô hình bóc tách được đề xuất dựa trên Tree DOM này, chẳng hạn dựa và kích thước của các nhánh con (độ lớn về mặt nội dung chứa trong chúng), hoặc dựa vào các thuộc tính như

Phần A – Tài liệu kỹ thuật VietSpider

màu sắc, font, ... định dạng cho đoạn văn bản chứa trong nhánh đó (Tree Item). Một giải pháp an toàn hơn cho việc nhận biết các nhánh có chứa nội dung mà ta đáng quan tâm đó là dựa vào tên nhánh và chỉ số nhánh để truy hồi đến đúng nhánh con cuối cùng có chứa nội dung.

Như chúng ta đã biết, hầu hết các website hiện nay đều là web tương tác động, mỗi loại dữ liệu sẽ được đưa vào cùng một định dạng trang giống nhau. Chẳng hạn, nội dung của mỗi bài báo sẽ tương ứng với một định dạng HTML tương đồng về mặt cấu trúc. Sự sai lệch diễn ra không đáng kể và có biên độ nhỏ, chỉ cần quan sát kỹ ta sẽ nhận ra điều này. Hình dưới đây là một minh dụ:

The screenshot displays the homepage of Tuổi Trẻ Online. At the top left is the logo 'tuoi tre online' with the tagline 'LƯU UY LỊCH SỬ VÀ TÂM HỒ - CHỈ TIẾNG THỜI ĐẠI'. To the right is the slogan 'TỐC ĐỘ CỦA THÔNG TIN'. Below this is a navigation bar with links for 'Dẫn Tin Điện Tử', 'Diễn Đàn', and 'Thư Viện Luật'. A search bar is located on the right with the text 'Tìm kiếm' and a search icon. Below the navigation bar, there are three main columns. The left column is a vertical menu with categories like 'Trang Chính', 'Chính trị - Xã Hội', 'Người Việt xa quê', etc. The middle column features a news article titled 'Ban bí thư đồng ý tạm đình chỉ công tác ông Trần Quốc Trường' with a sub-headline 'TT - Xem xét đề nghị của Thủ tướng, Ban bí thư Trung ương Đảng đồng ý về việc tạm đình chỉ công tác đối với ông Trần Quốc Trường - phó tổng thanh tra Chính phủ.' and a brief summary. Below the article is a red-bordered box containing social media sharing icons for 'Bản in', 'Phản hồi', 'Gửi tôi', 'Về đầu trang', and 'Chọn ngày'. The right column is titled 'Vụ án' and contains several small images with captions, such as 'Hầm chui Vành Thành 2 sẽ... kún mãi mãi' and 'Tàu lặn do phụ tùng kém chất lượng?'. At the bottom of the page, there is a list of news items with dates, such as 'Cả độ bóng đá qua mạng: Hở sơ hở chuyển VKS - (10/07)'.

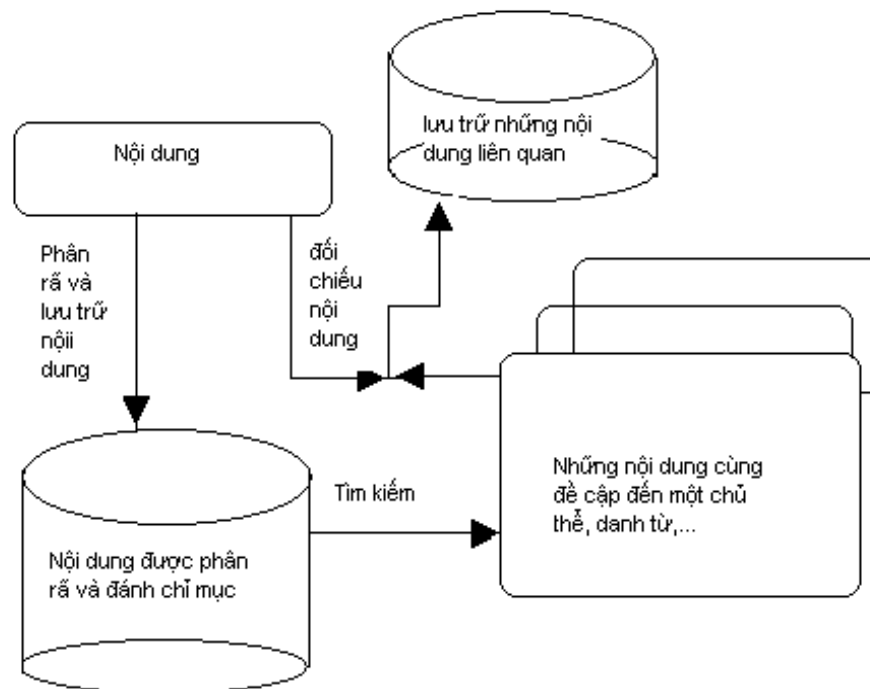
Bố cục giao diện của Website thể hiện ý đồ thiết kế của nhà thiết kế hoặc nhà phát triển Website đó. Những khoảng viền màu đỏ chỉ ra những phân vùng khác nhau trong một tin được phát hành trên trang Tuổi trẻ online. Nhìn vào hình, ta thấy Website là một đối tượng có “kiến trúc” thiết kế theo một khuôn mẫu nhất định được cố định chung cho mọi trang trong Website. Nội dung sẽ được đặt ở vùng trung tâm và những vùng khác sẽ giữ nguyên vị trí tương đối của chúng cho dù “chất và lượng” của nội dung có thể thay đổi. Sự ổn định mô hình dàn trang này là căn cứ tiến hành những kỹ thuật bóc tách an toàn và hiệu quả. Hình ảnh dưới đây sẽ cho chúng ta thấy việc truy xuất vào nội dung này thông qua Tree DOM – cấu trúc của Website được thể hiện qua dạng cây với các thẻ.

Phần A – Tài liệu kỹ thuật VietSpider

Data Mining là một lĩnh vực phân tích và tổng hợp ý nghĩa của một tập hợp dữ liệu. Trong những mô hình lý thuyết được trình bày, chúng ta thường tập trung vào giải quyết những bài toán cơ bản của data mining như: khai phá chủ đề của nội dung (Topic Distillation, Topic Extraction,...); phân loại nội dung (Classify Text, Text Category,...); tìm kiếm nội dung liên quan (Relation Learning);... Tùy mức độ yêu cầu, hệ thống khai thác thông tin có thể cài đặt Data Mining theo những dạng cụ thể khác nhau. Thông thường, chúng ta tập trung vào kỹ thuật phân loại nội dung theo ngữ nghĩa và tổng hợp những nội dung có liên quan ngữ nghĩa với nhau.

Về kỹ thuật phân loại, chúng ta thường định nghĩa trước một tập dữ liệu cho một thể loại nhất định và dùng nó để đối chiếu nội dung cần phân loại (còn gọi là từ điển phân loại). Hiện tại ứng dụng VietSpider chưa cài đặt được khả năng này, chúng tôi tập trung chủ yếu cho lĩnh vực thứ 2 là Relation Learning – tìm kiếm những nội dung liên quan.

Một bài báo thường đề cập đến một địa danh, một sự kiện, một thời điểm, con người hay những danh từ cụ thể, ... do đó dựa vào kỹ thuật phân tách nội dung, chúng ta có thể lấy ra những chủ thể được đề cập trong đó để thanh lọc tập nội dung liên quan. Tuy nhiên, việc lọc ra những chủ thể này không hẳn lúc nào cũng đưa ra các bài viết liên quan, những danh từ quá chung như tên một thành phố, tên một nước được đề cập trong bài báo chưa hẳn đã có sự liên quan về mặt ngữ nghĩa trong nội dung. Do đó, một bước thứ hai trong kỹ thuật Relation Learning là xem xét những sự việc cụ thể diễn ra ở nội dung, chẳng hạn : chống tham nhũng, chống dịch cúm gia cầm, yêu đương,... Khảo sát kết quả khi kết hợp hai bước kỹ thuật trên, chúng ta có được một mức độ chính xác tương đối cao và ổn định. Đây là những cài đặt cụ thể của hai mô hình thuật toán TF*PDF (Term Frequency * Proportional Document Frequency), Center Algorithm using LORs (Linked Object Representations) với sự hỗ trợ của kỹ thuật Stopping trong phân tách nội dung. Trong quá trình xây dựng, chúng tôi đã ứng dụng giải pháp đánh chỉ mục (indexing) và tìm kiếm (searching) nổi tiếng của Apache là Lucene Search. Đây là thư viện nguồn mở cho phép tìm kiếm nội dung văn bản đơn thuần trên một tập dữ liệu đã được đánh chỉ mục. Lucene Search được cài đặt để lưu trữ những nội dung đã được phân tích và tìm kiếm những nội dung cùng đề cập đến một danh từ, chủ thể cụ thể. Nó không phải là cốt lõi tạo dựng nên chức năng mining của ứng dụng nhưng là thành phần hỗ trợ đặc lực cho module này. Dưới đây là mô hình Relation Learning:



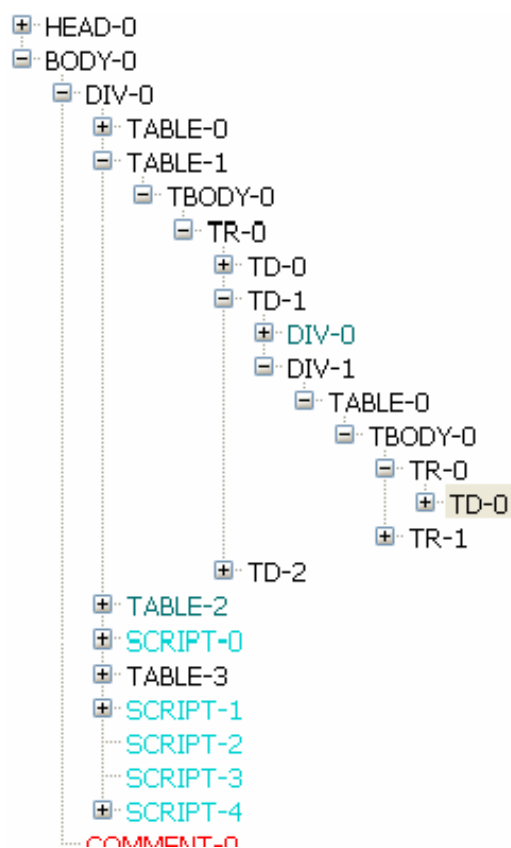
Đầu tiên, nội dung sẽ được phân tách bằng kỹ thuật Stopping, kỹ thuật này phân nội dung ra làm hai tập hợp chính: những chủ thể, danh từ được đề cập và sự việc, hành động diễn ra trong nội dung. Dữ liệu đó sẽ được lưu trữ trong database dưới dạng chỉ mục, bước tiếp theo là tìm kiếm ra những nội dung có cùng chủ thể, danh từ tương tự. Những nội dung đó được đem đổi chiều về ngữ nghĩa với nhau theo tiêu chuẩn cụ thể để chọn ra những nội dung thực sự liên quan với nhau. Sau đó, kết quả sẽ được lưu vào một hệ thống database khác.

IV. Những thành phần cơ bản của chương trình.

1. Bộ HTMLParser và công cụ HTML Explorer.

Trước khi tiến hành bóc tách, những nội dung dạng HTML text đơn thuần sẽ được chuyển sang một mô hình có giao diện lập trình (API) là DOM. Từ Tree DOM này, ta có thể tiến hành tập lệnh thực thi một vài nhiệm vụ cơ bản như bóc tách dữ liệu, lấy các liên kết (HTML Link) tiếp theo để truy vấn những nội dung khác, loại bỏ hoặc thực thi các scripting,.... Việc chuyển đổi từ dạng text đơn thuần sang Tree DOM sẽ do một bộ chuyển đổi thực hiện thường được gọi là HTMLParser. Tuy chức năng cũng giống như bộ Parser của Mozilla, nhưng HTMLParser trong chương trình được xây dựng tập trung vào các định dạng cấu trúc tài liệu (dàn trang) hơn là những thể hiện đồ họa của HTML Document. Một vài khả năng nữa của bộ HTMLParser là khả năng detect các encoding thông dụng trên Web, module dựa trên một sản phẩm open source viết lại bằng Java từ Mozilla Detect phát hành ở Sourceforge.net là JCharset. Khả năng encode và decode các ký tự quí ước trong HTMLReference. Hình dưới minh họa cho một HTML Tree DOM.

Phần A – Tài liệu kỹ thuật VietSpider

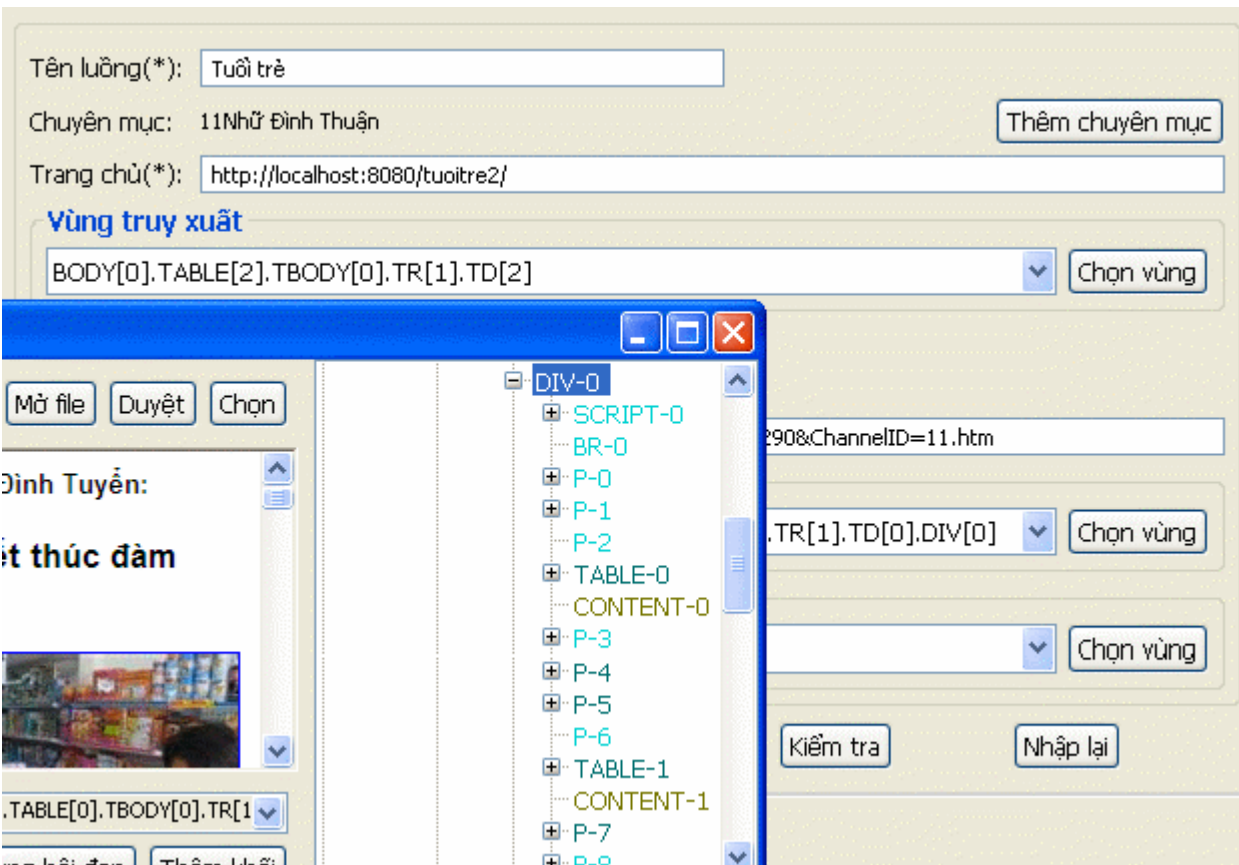


HTML Explorer là một công cụ có giao diện đồ họa cho phép truy vấn các thành phần của trang HTML trên một giao diện dạng cây – đó chính là HTML Tree DOM. Trong VietSpider, HTML Explorer là công cụ cho phép người dùng chỉ rõ phần nội dung cần bóc tách là phần nào trên trang. Khi cấu hình kênh tin mới, bạn cần nhập địa chỉ Website và nhấn nút chọn khối để bật giao diện chương trình.

2. Công cụ cấu hình một kênh khai thác thông tin.

Để có được mô hình khai thác và tổng hợp nội dung một cách toàn vẹn thì không thể chỉ dựa vào việc lấy lại những thông tin cơ bản như tiêu đề, mô tả nội dung,... mà phải có nội dung hoàn thiện. VietSpider cung cấp mô hình bóc tách một cách toàn vẹn, dĩ nhiên không thể thiếu công cụ định nghĩa kênh tin cho người sử dụng.

Phần A – Tài liệu kỹ thuật VietSpider

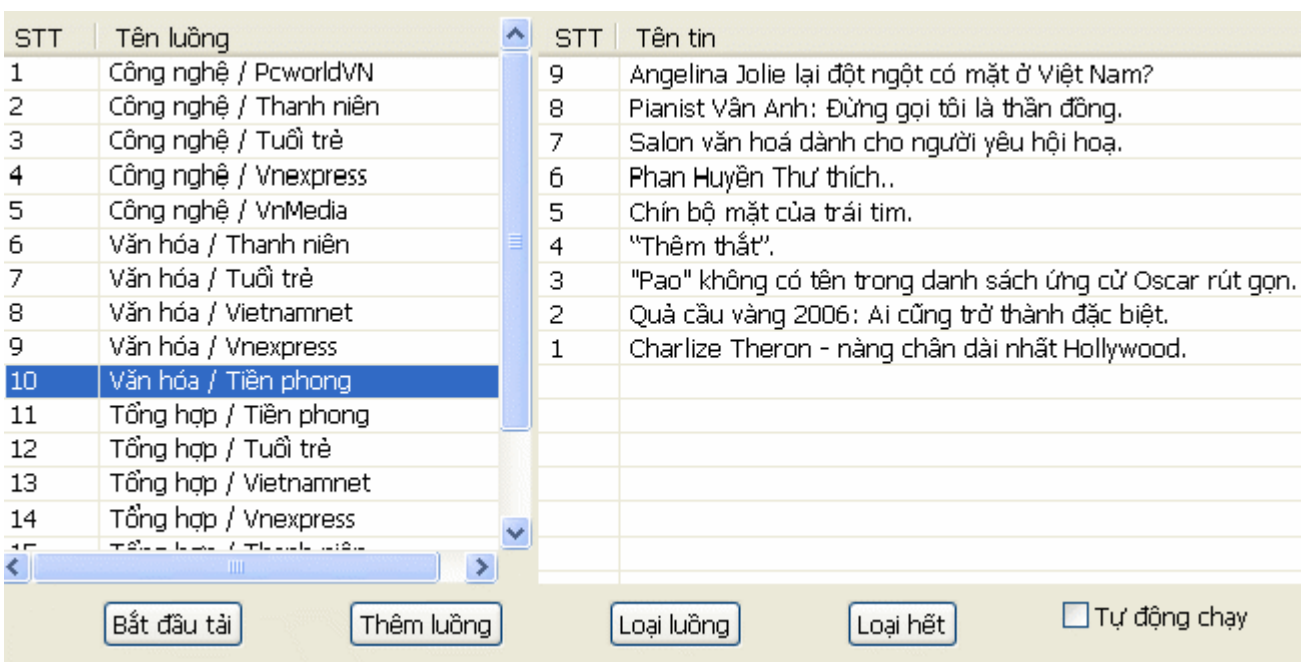


Công cụ cung cấp cách thức cho phép người sử dụng có thể cấu hình kênh khai thác nội dung bằng việc cung cấp một vài thông tin cơ bản (tên, trang chủ, định dạng, ...) và chỉ rõ vùng cần bóc tách dữ liệu. Mọi thao tác được thực hiện trên giao diện đồ họa, có thể bôi đen vùng nội dung cần bóc tách và chọn. HTMLExplorer cho phép truy vấn các đơn vị trong cấu trúc dữ liệu được định dạng từ nội dung bao gồm các thành phần của HTML Tree DOM thể hiện trên GUI. Thông tin được lưu dưới định dạng xml tổ chức theo mô hình tập tin, độc lập với cơ sở dữ liệu. Dựa vào dữ liệu này hệ thống Crawler có thể tiến hành bóc tách nội dung từ nguồn. Xin xem thêm hướng dẫn sử dụng trong tài liệu đi kèm.

3. Công cụ bóc tách và tổng hợp nội dung.

Module thứ hai trong giải pháp khai thác và tổng hợp nội dung là Crawler service. Đây là thành phần chính của chương trình. Đọc một số dữ liệu từ cấu hình, service truy vấn tới website nguồn, nhận và quét toàn bộ bề mặt với độ sâu được giới hạn trước đó rồi tiến hành bóc tách dữ liệu. Dữ liệu sau khi bóc tách được tổng hợp bằng việc trích rút những thông tin mô tả, lưu vào database. Dữ liệu trước khi lưu vào database đều là dữ liệu “sạch”, gọn bao gồm việc loại bỏ các scripting, định dạng css và đảm bảo sự chính xác về mặt cú pháp của trong định dạng html. Mọi thao tác quản lý trên giao diện đồ họa cho phép thêm mới, loại bỏ các kênh người sử dụng muốn bóc tách.

Phần A – Tài liệu kỹ thuật VietSpider



Một service khác đó là Mining Service chạy song song với Crawler service. Dịch vụ này sẽ đọc dữ liệu trong cơ sở dữ liệu và thực hiện việc đánh chỉ mục, tìm kiếm những nội dung liên quan. Cạnh đó là chức năng loại bỏ những dữ liệu cũ của chương trình để tránh việc lưu giữ những thông tin đã quá hạn làm chiếm dụng kích thước ổ cứng.

4. Giải pháp phát hành nhanh nội dung.

Nguồn dữ liệu được lưu trữ trong cơ sở dữ liệu, từ đó những nhà phát triển có thể tạo các trang tổng hợp tin tức bằng một quản trị nội dung đơn giản hay các phần mềm có nhu cầu sử dụng lại nội dung sau bóc tách. Tuy nhiên, phần mềm này có cung cấp chức năng phát hành nhanh nguồn nội dung đó trên giao diện đồ họa người dùng (GUI – Graphics User Interface) của chương trình hoặc thông qua website. VietSpider cũng đóng vai trò là một server của chương trình lắng nghe trên một cổng nhất định do quản trị hệ thống cung cấp, mọi truy vấn đến máy này thông qua trình duyệt sẽ được trả về một trang web báo chí thực thụ, từ đó có thể duyệt các nội dung khác một cách dễ dàng.

Chức năng tìm kiếm, lọc nội dung theo các từ khóa cung cấp, tổng hợp các sự kiện trong ngày, cho phép theo dõi các sự kiện xảy ra trong ngày, kênh RSS,...bổ sung thêm ngoài những tính năng cơ bản như duyệt và đọc nội dung.

V. Mô tả cơ sở dữ liệu cho chương trình.

Dưới đây là những mô tả cơ bản các bảng dữ liệu. Các kiểu dữ liệu có thể thay đổi tương ứng với database được lựa chọn cho phần mềm. Bảng mô tả dữ liệu này có thể có ích cho một số nhà quản trị muốn xây dựng lại website tổng hợp tin trên nền cơ sở dữ liệu của VietSpider.

1. Bảng Domain.

Là bảng chứa danh sách cách chuyên mục có dữ liệu được bóc tách trong ngày. Các bản ghi

Phần A – Tài liệu kỹ thuật VietSpider

bao gồm bốn thông tin cơ bản là ID - định danh duy nhất, Date – Ngày tháng cập nhật dữ liệu, Category – Chuyên mục chứa dữ liệu, Name – Tên nguồn dữ liệu. Truy vấn bảng này để có được những menu có chứa dữ liệu bóc tách. Chẳng hạn ngày 23-1-2007 bóc tách được các chuyên mục Văn hóa với các nguồn Tuổi trẻ, Vietnamnet; chuyên mục Thể thao với các nguồn VNExpress, Vietnamnet, Thanh niên; ngày 24-1-2007 bóc tách được các chuyên mục Văn hóa với các nguồn Thời báo Việt, Lao động, VnExpress; chuyên mục Thể thao với các nguồn Vietimes, Nhân dân, Thanh niên;...

Các bản ghi nôm nam sẽ là:

- 1 | 23-1-2007 | Văn hóa | Tuổi trẻ
- 2 | 23-1-2007 | Văn hóa | Vietnamnet
- 3 | 23-1-2007 | Thể thao | Vietnamnet
- 4 | 24-1-2007 | Văn hóa | Lao động

Nhìn vào một vài bản ghi minh dụ ta thấy số lượng chuyên mục có thể thay đổi theo từng ngày và số lượng kênh tin cũng thay đổi theo từng chuyên mục và có thể khác nhau. Do đó, thực đơn sinh ra trên giao diện theo lý thuyết sẽ không cố định như một số Website báo chí khác. Dưới đây là bảng mô tả các trường của bảng dữ liệu này:

Tên cột	Kiểu dữ liệu	Mô tả
ID	Numeric	Định danh duy nhất cho bản ghi dữ liệu
DATE	Text	Ngày tháng có dữ liệu được bóc tách và tổng hợp. Định dạng chữ từ VietSpider để tránh các định dạng ngày tháng với các hệ cơ sở dữ liệu khác nhau.
CATEGORY	Text	Tên chuyên mục có dữ liệu được bóc tách và tổng hợp. Có chứa ký tự unicode.
NAME	Text	Tên nguồn tin có dữ liệu được bóc tách và tổng hợp. Có chứa ký tự unicode.

2. Bảng Meta.

Bảng này làm nhiệm vụ lưu các dữ liệu đặc tả. Điều đó có nghĩa là nó chỉ nêu một số thông tin cơ bản của tập tin mà không lưu toàn bộ nội dung tập tin. Bản này có quan hệ với bảng Domain thông qua Domain.ID. Dữ liệu đặc tả về nội dung gần như được sinh tự động bởi chương trình.

Tên cột	Kiểu dữ liệu	Mô tả
ID	Numeric	Định danh duy nhất cho bản ghi

Phần A – Tài liệu kỹ thuật VietSpider

		dữ liệu
DOMAIN_ID	Numeric	Định danh quan hệ với bảng Domain dùng để chỉ rõ bản ghi nằm trong chuyên mục nào với ngày tháng và tên nguồn cụ thể.
TIME	Date	Thời điểm bóc tách nội dung.
SOURCE_TIME	Text	Ngày giờ mà website nguồn cập nhật dữ liệu. Dữ liệu trường này có thể có không có.
TITLE	Text	Tên nội dung. Có chứa ký tự unicode.
DES	Text	Mô tả hoặc tóm tắt nội dung. Có chứa ký tự unicode.
IMAGE	Text	Tên một ảnh nhỏ để minh họa cho nội dung.
URL	Text	Địa chỉ web cụ thể của nội dung bóc tách. Chẳng hạn http://vnexpress.net/Vietnam/Va-n-hoa/2007/08/3B9F98C7/

3. Bảng Content.

Bảng này chứa nội dung thực của tin tức. Bản này quan hệ với bảng Meta thông qua Meta ID.

Tên cột	Kiểu dữ liệu	Mô tả
META_ID	Numeric	Định danh quan hệ với bản Meta.
DATE	Text	Ngày nội dung được bóc tách. Hiện tại chỉ dùng cho hệ thống indexing.
CONTENT	Text	Nội dung của tin tức. Có chứa ký tự unicode và có thể dài.
STATUS	Numeric	Dùng cho hệ thống indexing ở các bản VietSpider trước build 9. Hiện tại bản build 9 chưa dùng đến.

4. Bảng Relation.

Bảng Relation thông tin các nội dung có liên quan tới nhau. Bảng này được cập nhật tự động bởi hệ thống mining và indexing.

Tên cột	Kiểu dữ liệu	Mô tả
META_ID	Numeric	Meta ID của nội dung có tin liên quan. Quan hệ với bảng Meta thông quan ID này.
RELATION_ID	Numeric	Meta ID của nội dung liên quan đến nội dung có Meta ID ở cột trên (Meta ID của nội dung bị liên quan).
PERCENT	Numeric	Mức độ liên quan của 2 nội dung với nhau. Mức độ này là quy định của hệ thống, không chính xác nhưng có thể dùng để đánh giá các tin là giống nhau hoặc liên quan với nhau theo một tiêu chí nào đó.

5. Bảng Image.

Bảng Image lưu các ảnh trong một nội dung. Bản này quan hệ với bảng Meta thông quan Meta ID. Các ảnh có thể được lưu trực tiếp trong bản này hoặc được lưu dưới dạng tập tin trong một định dạng thư mục của VietSpider. Khi đó ảnh sẽ được lấy ra theo tên ảnh đã lưu trong bản này.

Tên cột	Kiểu dữ liệu	Mô tả
ID	Numeric	Thứ tự của ảnh cho một nội dung với một hoặc nhiều ảnh (1,2,3,...). Chúng không là định danh duy nhất cho một bản ghi.
META_ID	Numeric	Định danh quan hệ với bản Meta chỉ rõ ảnh sẽ thuộc nội dung nào?
CONTENT_TYPE	Text	Kiểu của ảnh, chẳng hạn JPEG, GIF, BMP,...
NAME	Text	Tên của ảnh, lấy từ Website nguồn.

Phần A – Tài liệu kỹ thuật VietSpider

IMAGE	Binary	Ảnh thực – dữ liệu ảnh.
-------	--------	-------------------------

Trên đây là các bảng dữ liệu thiết kế cho việc lưu trữ nội dung sau bóc tách. Dưới đây là một bảng dữ liệu đề xuất thêm để lưu bộ lọc nội dung, chúng có thể có hoặc không. Nếu không có bảng này, chức năng tạo bộ lọc nội dung của VietSpider sẽ báo lỗi.

6. Bảng Filter.

Bảng Filter, bảng chứa tên và những từ khóa để lọc nội dung trong kho nội dung sau bóc tách. Chức năng lọc nội dung thực chất là khả năng tìm kiếm nội dung với một lượng từ khóa cho trước có chứa trong nội dung.

Tên cột	Kiểu dữ liệu	Mô tả
NAME	Text	Tên của bộ lọc
FILTER	Text	Các từ dùng để lọc nội dung, cách nhau dấu phẩy, không có khoảng trắng giữa hai đầu của từ.
META_ID	Text	Chứa một dãy các Meta ID của nội dung nối với nhau bằng dấu chấm phẩy, chúng là kết quả tạm thời của bộ lọc.

Chú ý: Mô hình cơ sở dữ liệu trên đây chỉ là mô hình cơ sở dữ liệu cơ bản cho VietSpider. Bạn có thể thiết kế một cơ sở dữ liệu khác có chứa các dữ liệu cơ bản này của VietSpider nhằm phục vụ cho hệ thống mình muốn xây dựng. Điều đó đồng nghĩa với việc tên các bảng, tên và số lượng các trường trong bảng có thể tùy ý thay đổi. Điều quan trọng nhất là bạn phải cập nhật các câu lệnh truy vấn dùng trong chương trình. Liên hệ với tác giả để nhận được sự hỗ trợ. Kiểu dữ liệu trong các bảng trên có thể tùy biến với từng hệ quản trị cơ sở dữ liệu khác nhau.

VI. Ý nghĩa thư mục dữ liệu.

Để quản trị VietSpider hiệu quả, bạn cần hiểu sơ lược về ý nghĩa một số thư mục trong chương trình. Phần dữ liệu của VietSpider nằm trong *data* của thư mục chương trình, các thư viện trong thư mục *lib*, các tập tin thực thi nằm trong thư mục chương trình.

Trong thư mục dữ liệu *data*, chúng ta có 4 thư mục con là *system*, *content*, *track*, *sources*.

Sources chứa các hình kênh khai thác thông tin được tổ chức theo từng chuyên mục khác nhau. *System* chứa: cấu hình cơ sở dữ liệu (bao gồm các câu lệnh sql để chương trình làm việc là *database.xml*, *dbload.xml*, *dbdelete.xml*, *dbsave.xml*). Tập tin *system.properties* chứa các cấu hình cho ứng dụng (dữ liệu được cập nhật trong phần cấu hình trên giao diện, bạn cũng có thể sửa đổi bằng tay trực tiếp trong file này và chạy lại ứng dụng để thiết lập các thông số). Tập tin *clean.properties* chứa các thông số sửa đổi nội dung sau bóc tách. Tập tin *datatime.cfg* chứa định dạng ngày tháng thông dụng trên các website nguồn. Thư mục *cms* chứa cấu mẫu cho

Phần A – Tài liệu kỹ thuật VietSpider

việc tạo trang phát hành nội dung.

Content chứa images dùng để lưu ảnh khi ảnh không lưu vào cơ sở dữ liệu. Ảnh sẽ được tổ chức vào các thư mục theo ngày. Thư mục indexed để chứa kho dữ liệu dùng cho hệ thống mining. Thư mục stores chứa các nội dung mà bạn muốn lưu trữ lại. Thư mục html là thư mục chứa nội dung sau bóc tách nếu bạn không muốn chứa vào database.

Track bao gồm: Thư mục downloaded lưu trữ những địa chỉ nội dung đã được lấy về. Thư mục logs ghi lại các lỗi của chương trình. Khi chương trình xảy ra lỗi, bạn có thể mở các tập tin (tên file đặt theo ngày tháng) và xem xét các thông tin về lỗi (nếu không xử lý được, xin hãy gửi các thông tin lỗi cho tác giả qua email nhudinhtuan@yahoo.com). Thư mục reporters chứa kết quả bóc tách theo ngày tháng và luồng bao gồm bao nhiêu nội dung được lấy về. Thư mục này sẽ ghi lại các kết quả để quản trị theo dõi việc bóc tách hàng ngày của Crawler và những kênh nào không lấy được nội dung rồi từ đó có những điều chỉnh cấu hình cho phù hợp. Thư mục test ghi lại các kết quả kiểm tra cấu hình hàng loạt các kênh trong danh sách tải. Khi bạn thực hiện việc kiểm tra các cấu hình, thông tin bóc tách sẽ được lưu vào các tập tin với tên tương ứng với chuyên mục và nguồn. Chức năng kiểm tra hàng loạt các cấu hình giúp bạn theo dõi những cấu hình nào đã bị sai so với website nguồn. Thư mục history dùng cho trình duyệt của VietSpider.

1. Làm sạch dữ liệu sau bóc tách.

Bạn có thể khai báo một số thông số để chương trình thực hiện việc làm sạch nội dung bóc tách được trước khi lưu vào cơ sở dữ liệu. Để có thể thực hiện được chức năng này, bạn cần hiểu qua về ngữ pháp HTML. Các khai báo nằm trong tập tin data/system/clean.properties. Ngữ pháp mẫu khai báo:

key=attribute|xóa hoặc không

Key: là tên thẻ. Chẳng hạn tôi viết, font=size thì có nghĩa là khi gặp các thẻ nào tên là font, xóa hết các thuộc tính là size đi. Viết font=size,face thì xóa hết các thuộc tính có tên là size hoặc face đi. Chẳng hạn một đoạn HTML như sau:

```
<font size="3" face="Arial"> Xin chào <font>. Khi làm sạch dữ liệu sẽ trở thành  
<font> Xin chào <font>.
```

Nếu key bằng *, chẳng hạn *=class,onclick thì gặp bất cứ thẻ HTML nào có các thuộc tính tên là class hoặc onclick là xóa các thuộc tính này đi.

Nếu viết style=true thì có nghĩa là tất cả phần nội dung nằm trong thẻ style sẽ bị xóa hết. Chẳng hạn một đoạn HTML sau:

```
<head>  
<title> Hello </title>  
<style>  
@page { size: 8.5in 11in; margin: 0.79in }  
P { margin-bottom: 0.08in }
```

```
</style>
```

```
<META NAME="GENERATOR" CONTENT="OpenOffice.org 2.0 (Win32)">
```

Phần A – Tài liệu kỹ thuật VietSpider

```
</head>
```

sau khi làm sạch sẽ là:

```
<head>
```

```
<title> Hello </title>
```

```
<META NAME="GENERATOR" CONTENT="OpenOffice.org 2.0 (Win32)">
```

```
</head>
```

nếu viết style=false thì sau khi làm sạch sẽ là:

```
<head>
```

```
<title> Hello </title>
```

```
<!--
```

```
@page { size: 8.5in 11in; margin: 0.79in }
```

```
  P { margin-bottom: 0.08in }
```

```
-->
```

```
<META NAME="GENERATOR" CONTENT="OpenOffice.org 2.0 (Win32)">
```

```
</head>
```

Như vậy nếu viết style=false thì phần nội dung trong thẻ style sẽ bị đẩy thành comment chứ không loại bỏ.

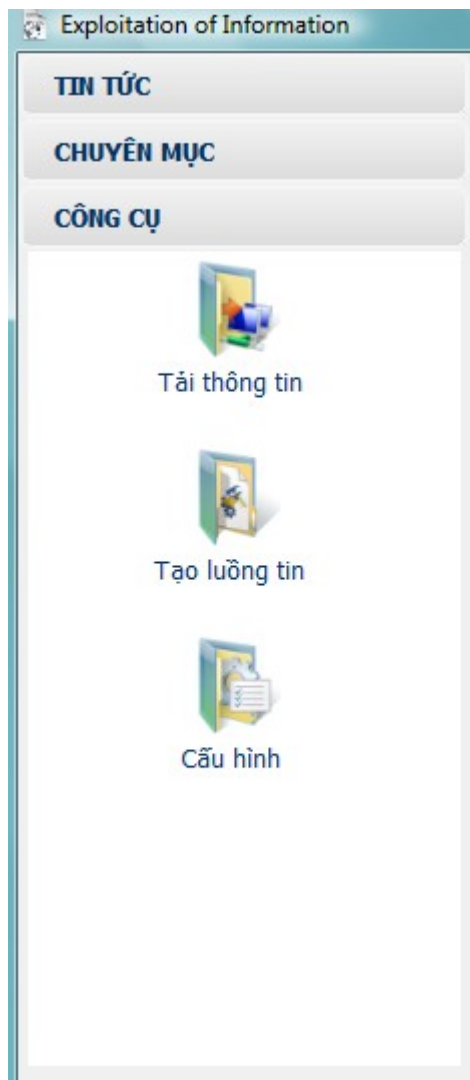
Ví dụ nếu tôi viết h1=false khi đó phần nội dung trong h1 sẽ không được định dạng là h1 nữa mà sẽ trở thành đoạn chữ bình thường. Nếu tôi viết h1=true thì phần nội dung đó sẽ bị loại bỏ ra khỏi nội dung sau bóc tách.

Trên đây là một vài ngữ pháp cơ bản mà bạn có thể khai báo để Crawler của VietSpider làm sạch dữ liệu ở dạng thô cho bạn.

2. Cấu hình một số thông số cho chương trình.

Các thông số cấu hình sẽ được lưu vào tập tin data/system/system.properties. Bạn có thể trực tiếp mở tập tin này ra sửa đổi hoặc dùng giao diện đồ họa người dùng để thao tác. Bạn chọn **Công cụ -> Cấu hình**.

Phần A – Tài liệu kỹ thuật VietSpider



Cửa sổ cấu hình sẽ xuất hiện như hình dưới đây.

Phần A – Tài liệu kỹ thuật VietSpider

The screenshot shows the VietSpider configuration interface. It is divided into several sections:

- Truy xuất tới máy:** A text box containing "http://thuan:9245/".
- Địa chỉ máy:** An empty text box.
- Cổng ứng dụng:** A text box containing "9247".
- Tên ứng dụng:** A text box containing "vietspider".
- Update site:** An empty text box.
- Thư mục sao lưu:** A button next to a text box containing "C:\Temp".
- Bộ nhớ ứng dụng (mb) từ:** A spinner box set to "64".
- tới:** A spinner box set to "320".
- Đặt Proxy:** A section with fields for "Địa chỉ:", "Cổng:", "Loại proxy:" (set to "NONE"), "Tên:", and "Mật khẩu:".
- Cấu hình Crawl:** A section with various settings:
 - Thời gian nghỉ (phút): 30
 - Thời gian lưu dữ liệu (ngày): 5
 - Thời điểm (giờ) xóa dữ liệu: 10
 - Mức độ tin liên quan tối thiểu(%): 10
 - Giới hạn nội dung tải (ngày): 1
 - Tự động chạy
 - Tải ảnh
 - Bắt buộc tải ảnh
 - Bỏ tên
 - Bỏ mô tả
 - Lưu nội dung ra thư mục
 - Lưu ảnh ra thư mục
 - Làm sạch dữ liệu
- Lưu:** A button at the bottom right.

Xin được giải thích các thành phần trong cửa sổ cấu hình này.

- **Truy xuất tới máy:** Khai báo địa chỉ và cổng để truy xuất tới một máy khác thông qua HTTP. Nhờ việc truy xuất tới máy này mà bạn có thể tiến hành một số những trao đổi như trao đổi cấu hình các kênh tin. Cập nhật cấu hình các kênh tin từ máy bạn. Các phiên bản tương lai có thể làm nhiều nhiệm vụ khác nữa thông qua giao tiếp này.
- **Địa chỉ máy:** Khai báo địa chỉ máy để Server dựng lên. Thông thường bạn không cần phải khai báo ở đây, chương trình sẽ tự động lấy tên máy làm tên server, khi đó, các máy khác trong mạng LAB có thể truy cập tới máy này để đọc nội dung xuất bạn qua Web bằng trình duyệt hoặc tiến hành trao đổi dữ liệu thông qua tính năng **Truy xuất tới máy** như đề cập ở trên.
- **Cổng ứng dụng:** mặc định, chương trình sẽ lắng nghe trên cổng 9245, bạn có thể đặt lại cổng này cho việc lắng nghe các yêu cầu truy vấn tới chương trình. Chẳng hạn, khi VietSpider trên máy bạn được chạy, người ở máy khác có thể gõ vào http://tên máy bạn:9245 trong ô địa chỉ của trình duyệt và họ có thể duyệt nội dung mà VietSpider của bạn đã bóc tách. Cổng và địa chỉ máy khai báo này cũng được dùng cho việc quản trị nội dung trong chính VietSpider của bạn.
- **Tên ứng dụng:** Không quan trọng, đặt lại tên ứng dụng mà bạn mong muốn để chúng có thể thể hiện là một thành phần ở các url truy vấn tới nội dung.
- **Backup Folder:** Thư mục sao lưu dữ liệu trước khi dữ liệu được xóa khỏi database.
- **Update site:** Địa chỉ cập nhật chương trình, hiện tại chưa chạy tốt.
- **Đặt Proxy:** Nếu bạn connect tới Internet thông qua một Proxy, bạn cần khai báo các thông số Proxy này để chương trình có thể tải được tin tức.
- **Thời gian nghỉ:** Thông thường Crawler sẽ nhận vào một danh sách luồng tải và tiến hành quét từ đầu tới cuối, sau khi quét xong một lượt, chương trình sẽ tự động nghỉ một khoảng thời gian nhất định rồi tiến hành quét lượt tiếp theo. Đặt lại giá trị này tùy theo ý bạn muốn.

Phần A – Tài liệu kỹ thuật VietSpider

- **Thời gian lưu dữ liệu:** Là khoảng thời gian mà nội dung có thể lưu trong cơ sở dữ liệu của bạn tính từ ngày hiện tại. Dữ liệu của những ngày sau đó sẽ bị xóa đi, nếu **Backup Folder** được khai báo, dữ liệu sẽ được sao lưu theo ngày ra thư mục đó trước khi xóa.
- **Thời điểm xóa dữ liệu:** Đơn vị là giờ, là lúc chương trình thực hiện việc kiểm tra dữ liệu đã quá hạn, tiến hành xóa và sao lưu ra thư mục trước khi xóa dữ liệu.
- **Bộ nhớ tối thiểu:** Khai báo bộ nhớ tối thiểu cho chương trình. Khởi động Spider.exe thì việc đặt lại này mới có tác dụng.
- **Bộ nhớ tối đa:** Khai báo bộ nhớ tối đa cho chương trình. Khởi động Spider.exe thì việc đặt lại này mới có tác dụng.
- **Mức độ tin liên quan:** Đặt lại mức độ phần trăm liên quan giữa các nội dung với nhau, nếu hệ thống đánh giá dưới mức này nó sẽ không lưu lại cơ sở dữ liệu. Nếu những nội dung liên quan với nhau bằng hoặc vượt qua con số phần trăm này, nó sẽ lưu lại trong cơ sở dữ liệu.
- **Giới hạn nội dung tải:** VietSpider tự động nhận biết ngày giờ nội dung được cập nhật từ website nguồn (nếu có và nằm trong vùng bóc tách). Sau đó nó so sánh thời gian, nếu những nội dung quá cũ với khoảng thời gian bạn chọn ở đây thì sẽ không được lưu lại trong cơ sở dữ liệu. Nếu bạn chọn là 0 ngày thì tất cả các nội dung bóc tách được sẽ lưu vào cơ sở dữ liệu.
- **Tự động chạy:** Tự động chạy Crawler mỗi khi chạy VietSpider, theo đó chương trình sẽ tiến hành bóc tách ngay từ lúc khởi động VietSpider, bằng không, bạn phải vào **Tải thông tin** để chạy Crawler.
- **Tải ảnh:** Cho phép tải ảnh về hay không, nếu không, nội dung sẽ lưu đầy đủ url của ảnh.
- **Bỏ tên:** Bỏ tên ra khỏi nội dung.
- **Bỏ mô tả:** Bỏ phần trích rút từ nội dung ra làm mô tả ra khỏi nội dung.
- **Lưu nội dung ra thư mục:** Lưu nội dung ra thư mục trong chương trình dưới dạng tập tin html, khi đó database sẽ nhẹ bớt.
- **Lưu ảnh ra thư mục:** Lưu ảnh ra thư mục dưới dạng tập tin, khi đó database sẽ nhẹ bớt.
- **Bắt buộc tải ảnh:** Nếu không chọn tính năng này, chương trình khi tải ảnh bị lỗi (do đường truyền bị lỗi) thì nội dung vẫn được lưu lại, ảnh bị lỗi sẽ chỉ là một url đầy đủ tới website nguồn. Bằng không, chương trình sẽ không lưu nội dung đó, việc tải và bóc tách nội dung đó sẽ tiếp tục được tiến hành ở lần quét sau.
- **Làm sạch dữ liệu:** Dữ liệu sau bóc tách có thể loại bỏ một số định dạng như màu mè, phong chữ, kiểu chữ,... Việc loại bỏ những định dạng đó sẽ được khai báo trong tập tin `clean.properties` ở *thư mục chương trình/data/system*. Chúng tôi sẽ có những ghi chú cho việc khai báo các loại bỏ này ở tài liệu chi tiết sau.

Sau khi thực hiện các khai báo hoàn chỉnh về cài đặt chương trình, nhấn nút **Lưu** và khởi động lại chương trình. Nhấp chuột vào hai mũi tên hướng xuống trong góc phải phía trên và chọn **Thoát** sau đó chạy lại chương trình.

Bây giờ, bạn có thể khai thác thông tin qua VietSpider.

VII. Tài liệu và địa chỉ tham khảo.

1. Extracting Tourism Information from the Web.

Informationssysteme Abteilung für Datenbanken und Artificial Intelligence der Technischen

Universität Wien der Anleitung von Univ.Prof. Dipl.-Ing. Dr.techn. Georg Gottlob und.-Ing. techn. Marcus Herzog.

2. Semi-automatic Wrapper Generation for Information Sources .

Naveen Ashish and Craig Knoblock - Information Sciences Institute and of Computer Science University of Southern California AdmiraltyWay, Marina del Rey, CA 90292, <http://www.isi.edu/sims/naveen>.

3. Building Intelligent Systems for Mining Information Extraction Rules from Web Pages by using Domain Knowledge.

Heekyoung Seo - HCI Lab, Samsung Advanced Institute of Technology, KOREA @sait.samsung.co.kr; Yang and Joongmin Choi of Computer Science and Engineering, Hanyang University, KOREA.

4 .Visual Information Extraction.

Yonatan Aumann, Ronen Feldman, Yair Liberzon, Benjamin Rosenfeld, Jonathan Schler - Department of Computer Science, Bar Ilan University, Ramat Gan 52900, Israel ClearForest Ltd., 6 Yoni Netanyahu St., Or Yehuda 60376, Israel.

5. XWRAPComposer: A Multi-Page Data Extraction Service Bio-Computing Applications.

Ling Liu, Jianjun Zhang, Wei Han, Calton Pu, James Caverlee, Sungkeun Park - College of Computing, Georgia Institute of Technology Terence Critchlow, Matthew Coleman, David Buttler
Lawrence Livermore National Laboratory, California, USA.

6. Adaptive Semi-structured Information Extraction.

Anders Arpteg - The Computer and Information Science Department Linköping university, SE-581 83, Linköping, Sweden, <http://www.ida.liu.se/>.

7. Visual Wrapping and Functional Linkage of Existing Web Applications.

Kimihito Ito, Yuzuru Tanaka - Meme Media Laboratory, Hokkaido University N-13 W-8, Sapporo, 060-8628, JAPAN.

8. Coordinating Heterogeneous Web Services through Handhelds using SyD's Wrapper Framework.

Dr. Sushil K. Prasad - Dr. Anu G. Bourgeois - Dr. Alex Zelikovsky - Presented in partial fulfillment of requirements for the Degree of Master of Science in the College of Arts and Sciences, Georgia State University.

9. VIPs : a Vision-based Page Segmentation Algorithm.

Deng Cai, Shipeng Yu, Ji-Rong Wen, Wei-Ying Ma from Microsoft Research.

10. Statistical Language Modeling For Information Retrieval.

Xiaoyong Liu and W. Bruce Croft - Center for Intelligent Information Retrieval - Department of Computer Science University of Massachusetts, Amherst, MA 01003.

11. Word Pairs in Language Modeling for Information Retrieval.

Carmen Alvarez, Philippe Langlais and Jian-Yun Nie - Dept. IRO, Université de Montréal CP. 6128, succursale Centre-ville Montréal, Québec, H3C CJ7 Canada.

12. Variations on Language Modeling for Information Retrieval.

Wessel Kraaij - Enschede: Neslia Paniculata. Enschede - With ref. With summary 90-75296-09-6 ISSN 1381-3617; No. 04-62 (CTIT Ph.D. -thesis series) - Wessel Kraaij.

13. A Significant Improvement to Clever Algorithm in Hyperlinked Environment.

Minhua Wang - Department of Computer Science and Engineering State University of New York at Buffalo Buffalo, NY 14260 and College of Business and Information Systems Dakota State University.

14. Improved Algorithms for Topic Distillation in a Hyperlinked Environment.

Krishna Bharat and Monika R. Henzinger Digital Equipment Corporation Systems Research Center 130 Lytton Avenue Palo Alto, CA 94301.

15. Topic Extraction from News Archive Using TF*PDF Algorithm.

Khoo Khyou Bun and Mitsuru Ishizuka - Dept. of Information and Communication Engineering - The University of Tokyo.

16. Text Categorization and Relational Learning.

William W. Cohen - AT&T Bell Laboratories.

17. Extracting and Using Relationships found in Text for Topic Tracking.

Paul Ogilvie, Jame Allan, David Jensen, Walter Rosenkrantz.

18. A functionality taxonomy for document search engines.

Rik D.T. Janssen, Henderik A. Proper Institute 6 PV Gouda Netherlands. Proper@acm.org.

19. Structured Index System at NTCIR Workshop 2: Information Retrieval Methods Using Ordered Co-occurrence of Words and their Dependency Relationships.

Atsushi MATSUMURA, Atsuhiko TAKASU, Jun ADACHI - National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.

20. Providing Government Information on the Internet: Experiences with THOMAS.

W. Bruce Croft and Robert Cook - Computer Science Department University of Massachusetts Amherst, MA 01003-4610. Dean Wilder - Library of Congress ITS 9332, 101 Independence Avenue Washington, D.C. 20540.

21. Silk from a Sow's Ear: Extracting Usable Structures from the Web.

Peter Pirolli, James Pitkow, Ramana Rao - Xerox Palo Alto Research Center 3333 Coyote Hill Road Palo Alto, CA 94304, USA

22. Designing the User Interface: Strategies for Effective Computer Interaction.

B. Schneiderman. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1992.