HỢP NHẤT CÁC TẬP TIN CÓ CẤU TRÚC DẠNG XML

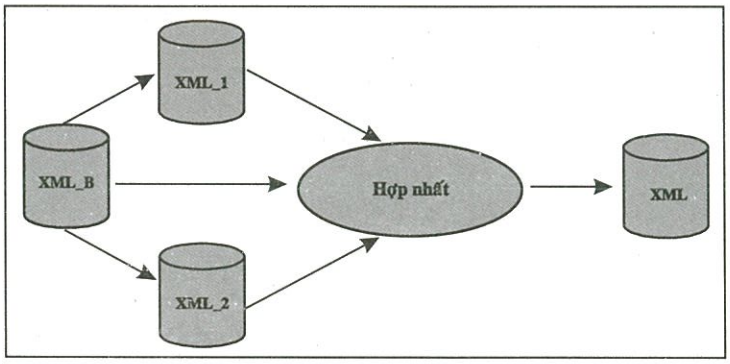
Đồng Thị Bích Thủy[1], Đinh Hùng[2]
[1] Khoa CNTT - Đại Học Khoa Học Tự Nhiên – ĐHQG-HCM
[2] Trung tâm Giải pháp Phần mềm – Công ty HPT VietNam
(Bài nhận ngày 12 tháng 05 năm 2003)

TÓM TẮT: *Bài báo sử dụng cách tiếp cận hợp nhất cấu trúc 3-way, lấy một tập tin làm cơ sở (Merging from a common ancestor) với thông tin đầu vào là ba tập tin XML T_B, T_1, T_2 với T_B là tập tin cơ sở, các tập tin T_1, T_2 là tập tin nhánh có thể chứa các thay đổi so với tập tin cơ sở T_B.*
Quá trình hợp nhất đặt cơ sở trên việc xác định ánh xạ cấu trúc giữa các cây con trong các tập tin tham gia. Để tăng tính chính xác của ánh xạ, bài báo đề nghị cách tinh chỉnh ánh xạ này bằng việc áp dụng từ điển WordNet đối với các phần tử thẻ XML. Việc tìm kiếm ánh xạ này là bước cần thiết trước khi việc hợp nhất diễn ra. Ngoài ra, cách tiếp cận đề nghị của bài báo cho phép nhận biết và xử lý các đụng độ nảy sinh trong quá trình hợp nhất dữ liệu XML ở các mức độ khác nhau. Giải pháp xử lý các đụng độ này là chìa khóa để đạt được việc hợp nhất hiệu quả.

1. Giới thiệu bài toán

1.1. Bài toán hợp nhất

Bài báo đề nghị cách tiếp cận **hợp nhất cấu trúc 3-way** (Merging from a common ancestor), với thông tin đầu vào là ba tập tin XML T_B, T_1, T_2 với T_B là tập tin cơ sở, các tập tin T_1, T_2 là tập tin nhánh có thể chứa các thay đổi so với tập tin cơ sở T_B. Việc đồng bộ hóa sẽ dựa vào việc truyền một số tập tin trung gian đến các nguồn thay đổi, các tập tin trung gian này cho phép truyền đi chỉ phần khác biệt chứ không phải toàn bộ tập tin, vì thế sẽ tiết kiệm băng thông truyền tải. Ngoài ra, cách tiếp cận đề nghị của bài báo cho phép nhận biết và xử lý các đụng độ nảy sinh trong quá trình hợp nhất dữ liệu XML ở các mức độ khác nhau. Giải pháp xử lý các đụng độ này là chìa khóa để đạt được việc hợp nhất hiệu quả.

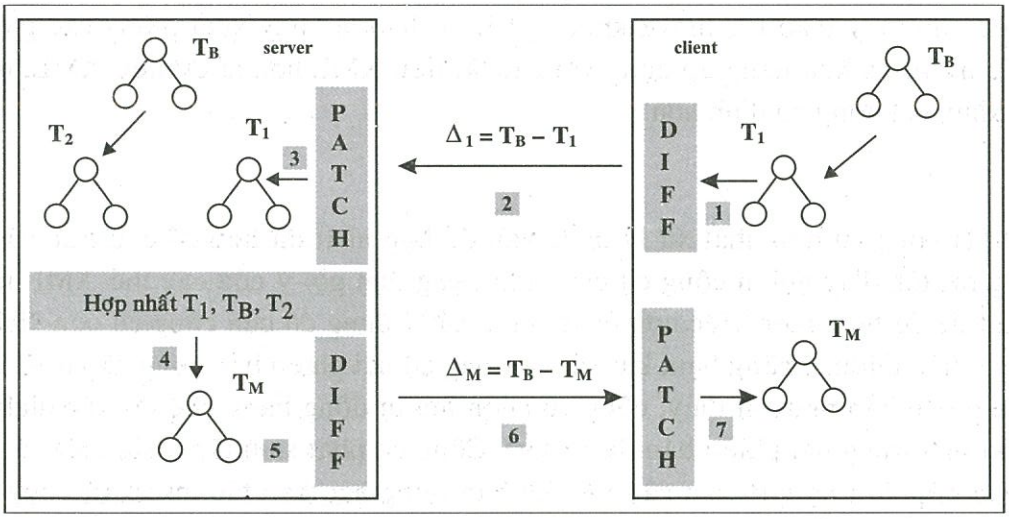


Hình 1. Cách hợp nhất 3-way

1.2 Bài toán đồng bộ hóa

Khi các máy tính từ xa (client) cần được đồng bộ hoá trong môi trường băng thông kém đến máy chủ (server). Hệ thống hợp nhất có thể được sử dụng để truyền các thông tin khác biệt (gọi là delta) và tổng hợp lại tập tin XML trong qui trình xử lý đồng bộ hóa.

1. Tạo delta: client tạo một tập tin delta để lưu tất cả các thay đổi được thực hiện (T_B-T_1 = Δ_1).
2. Gửi delta đến server: tập tin delta (thường nhỏ) được gửi đến server.
3. Tái tạo lại dữ liệu của client (tập tin T_1): server khi đó sẽ tạo lại trạng thái dữ liệu của client bằng cách ráp Δ_1 với T_B ($T_B + \Delta_1 = T_1$).
4. Hợp nhất: Tác vụ hợp nhất 3-way sẽ được thực hiện tại server để tổng hợp các chỉnh sửa được thực hiện tại client và tại server (T_B, T_1, T_2).
5. Tạo tập tin Δ_M: server sẽ tạo một tập tin delta mới từ tập tin hợp nhất T_M và T_B (T_B -T_M = Δ_M).
6. Tập tin Δ_M được gửi đến client và được ráp lại với T_B ($T_B + \Delta_M = T_M$).



Hình 2. Đồng bộ hoá dữ liệu bằng phương pháp hợp nhất 3-way.

1.3. Cách tiếp cận

Bài toán hợp nhất 3-way theo cấu trúc để đồng bộ hoá được giải quyết bằng cách chia nó thành ba bài toán cụ thể như sau.

Giả sử T_1 và T_2 là hai cây có thứ tự được dẫn xuất từ cây T_B. Chúng ta sẽ phân tích và thiết kế một công cụ có thể:

1. Thực hiện việc hợp nhất 3-way theo cấu trúc các cây T_1, T_2 và T_B và phát hiện, diễn tả mọi đụng độ xảy ra trong khi hợp nhất. Gọi là bài toán hợp nhất cây.
2. Sinh ra tập khác biệt giữa hai cây T_1 và T_2 dưới dạng một kịch bản chỉnh sửa. Sử dụng tập khác biệt và thông tin của cây T_1 nhận lại được cây T_2. Gọi là bài toán tạo khác biệt và ráp cây.

2. Hiện trạng

Trong mục này chúng ta sẽ xem xét một số cách tiếp cận hợp nhất theo cấu trúc.

UNIX diff3

Công cụ UNIX diff3 là đầu cuối (front-end) của công cụ diff, nó cung cấp hỗ trợ cho việc hợp nhất và tạo khác biệt các tập tin văn bản.

Giả sử chúng ta có tài liệu cơ sở B, nó có hai nhánh B_1 và B_2. Diff3 thực hiện việc hợp nhất bằng cách sinh ra tập khác biệt $B \Theta B_1$ và ráp vào B_2, hoặc bằng cách ráp vào B_1 tập khác biệt $B \Theta B_2$.

IBM Alphawork – XML Diff và Merge Tool (XMLDiff)

IBM's Alphaworks cũng phát triển một "công cụ hợp nhất và tạo khác biệt XML" (XMLDiff) [3]. Công cụ này so sánh hai tập tin đầu vào XML, và thể hiện các khác biệt. Người dùng khi đó sẽ có thể hợp nhất tương tác hai phiên bản bằng cách chọn nhánh này hay nhánh kia tại những vị trí khác nhau.

DeltaXML

Công cụ hợp nhất dùng để đồng bộ hóa, Trong DeltaXML [7] không xét đến thao tác move do họ cho rằng thảo luận với một tác giả phân tích các thay đổi trong 40000 tập tin XML trên Web cho thấy thao tác move không phải là thao tác hay xuất hiện. Tác giả cho rằng Thao tác move có khả năng áp dụng với các tài liệu XML hơn là dữ liệu XML, nơi mà cấu trúc có khuynh hướng cố định hơn.

3DM

3DM [8] là công cụ hợp nhất xử lý được vấn đề hợp nhất dữ liệu có cấu trúc và có cơ chế mở rộng rất tốt. Tuy nhiên công cụ chưa tận dụng tính gợi ý của các thẻ XML mà chỉ dựa vào cấu trúc để thực hiện việc hợp nhất. Việc xử lý đụng độ của công cụ còn khá chủ quan và không thân thiện. Chẳng hạn, khi xảy ra đụng độ hai phiên bản cùng cập nhật một hạt dữ liệu của phiên bản nguyên thủy, công cụ chọn lựa tự động bằng chế độ xác định quyền ưu tiên (cho một trong hai phiên bản dẫn xuất). Công cụ phát sinh tập khác biệt và ráp cây sử dụng trực tiếp ánh xạ giữa hai cây và giới hạn trong hai thao tác insert và copy không tạo được tập khác biệt tối ưu.

3. Giải quyết bài toán

3.1. Giống nhau về nội dung – ngữ nghĩa

Xác định độ giống nhau nội dung bằng q-gram

Khoảng cách chuỗi q-gram giữa hai chuỗi được định nghĩa như sau [5]:

Cho Σ là bộ ký tự hữu hạn, Σ^* là tập tất cả các chuỗi sinh ra từ Σ và Σq là tất cả các chuỗi có chiều dài q sinh ra từ Σ. Một q-gram là một chuỗi $v = a1a2...aq \in \Sigma q$.

Cho v là một q-gram và $x = a1a2...an$ là một chuỗi trong Σ^*. Nếu $v = aiai+1...ai+q-1 \subset x$ với i nào đó, thì v xuất hiện trong x. Chúng ta ký hiệu số lần xuất hiện của v trong x là Gq(x)[v]. q-gram profile của x là vectơ $Gq(x) = (G(x)[v]), v \in \Sigma q$.

Khoảng cách q-gram giữa các chuỗi x và y là chuẩn L1 của Gq(x) – Gq(y) : $Dq(x, y) = \Sigma_{v \in \Sigma^q} |G_q(x)[v] - G_q(y)[v]|$.

Xác định độ giống nhau ngữ nghĩa bằng từ điển Wordnet

WordNet [6] chứa thông tin chuẩn được tìm thấy trong các tự điển và tự điển đồng nghĩa.

Một khái niệm thường sử dụng trong WordNet là *hypernym*. Một *hypernym* của một thuật ngữ là một thuật ngữ tổng quát hơn phù hợp với phát biểu "_____ *is a kind of* ____". Ví dụ, dog là một loại canine, do đó canine là một hypernym của dog. Một canine là một loại mammal, do đó mammal là một *hypernym* của canine. Điều này cứ tiếp tục cho đến khi đạt đến một thuật ngữ đỉnh nào đó, rất tổng quát; trong trường hợp này, đỉnh hypernym của dog và canine là entity. Các thuật ngữ này, liên quan với nhau thông qua khái niệm *hypernymy*, hình thành một cây *hypernym*. Quan hệ ngược lại gọi là *hyponymy*.

*Khoảng cách giữa 2 nút:

Cho 2 nút A và B. Khoảng cách giữa A và B được ký hiệu là Distance(A, B) được tính bằng công thức:

$$\text{Distance}(A, A) = 0$$

$$\text{Distance}(A, B) = \begin{cases} 0 \text{ nếu Similar}(A, B) \\ \text{Distance}(A,C) + \text{Distance}(B,C) \text{ với } C = \text{NearestAncestor}(A, B) \\ \text{nếu not}(\text{Similar}(A, B)) \\ (A <> B) \end{cases}$$

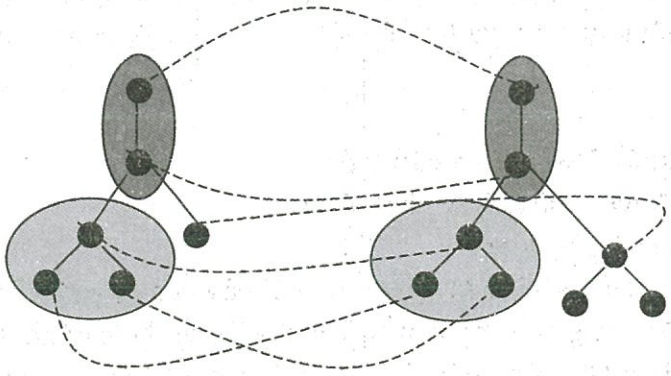
Xác định từ đồng nghĩa tiếng Việt

Hiện nay, từ điển WordNet tiếng Việt chưa được xây dựng, và việc hiện thực từ điển WordNet tiếng Việt nằm ngoài phạm vi của bài báo. Để minh họa khả năng sử dụng từ điển WordNet tiếng Việt trong tương lai, chúng ta xây dựng bộ từ điển đồng nghĩa (và phản nghĩa) tiếng Việt.

Chúng ta sử dụng từ điển tiếng Việt để đo hai từ đồng nghĩa (phản nghĩa) trong quá trình hợp nhất.

Giải thuật tìm kiếm ánh xạ cây con

Chúng ta sẽ tìm kiếm các cây con lớn nhất có thể có của T_B và T_I ánh xạ được với nhau bằng giải thuật heuristic greedy.



***Hình 3** Ánh xạ giữa hai cây*

Ánh xạ giữa 2 cây là tìm các cây con tương ứng nhau trong 2 cây. Sự tương ứng (ánh xạ) được định nghĩa như sau:

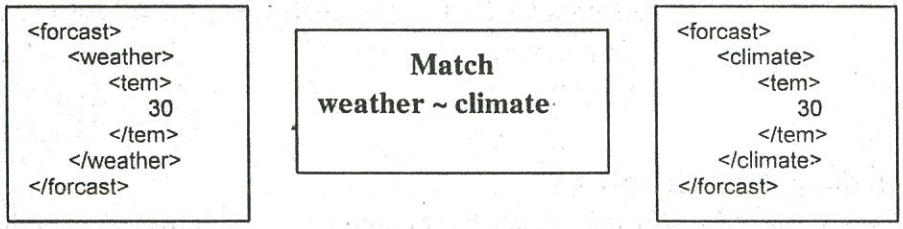
Ánh xạ chính xác giữa 2 cây con:
Nội dung của node gốc và các node con cũng như thứ tự các node con cũng phải giống nhau hoàn toàn. Nói chung có thể chồng khít 2 cây lên nhau được.

Ánh xạ tương đồng cấu trúc:
Nội dung nút gốc có thể khác nhau nhưng nội dung cũng như thứ tự của các node con phải giống nhau hoàn toàn.Vậy các chiến lược chấp nhận ánh xạ tương đồng sẽ phụ thuộc cách chấp nhận độ sai lệch giữa 2 nút gốc với nhau. Chính tại yếu tố này chúng ta sẽ can thiệp vào bằng 2 cách đo độ sai lệch đó là 2 phương pháp sử dụng:

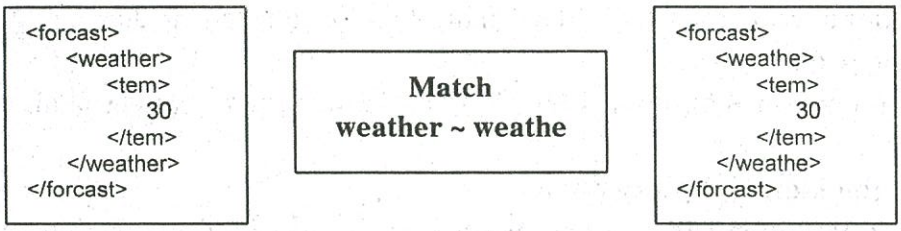
-Từ điển đồng nghĩa
-Q-gram

Sau đây là minh hoạ của các phương pháp trên:

*Từ điển đồng nghĩa



*Q-gram



Để thực hiện việc ánh xạ cây con ta sử dụng giải thuật greedy: lấy một node trong T_1 và toàn bộ cây T_B làm thông tin vào, và trả về một node trong T_B là gốc của cây con ánh xạ lớn nhất.

3.3. Thuật toán hợp nhất 3-way theo cấu trúc
Chúng ta có thể hình dung ý tưởng hợp nhất như sau:
Tiến trình hợp nhất gồm 3 giai đoạn:

* Giai đoạn tạo danh sách hợp nhất của mỗi cây con dựa vào cây T_B. Giai đoạn này các thay đổi đặc trưng của cây con so với cây T_B sẽ được đánh dấu bằng 2 thao tác là treo những node được thêm mới và khoá những node bị di chuyển.
* Giai đoạn tạo thành các cặp hợp nhất. Input của giai đoạn này chính là 2 danh sách hợp nhất mà ta có được từ giai đoạn 1 của 2 cây con T_1 và T_2. Output là danh sách các cặp node sẽ được thực hiện hợp nhất ở giai đoạn sau. Mọi thay đổi đặc trưng của 2 cây con mà ta xác định ở giai đoạn 1 sẽ được bảo lưu trong giai đoạn này.
* Giai đoạn hợp nhất các cặp lấy từ danh sách có được ở giai đoạn 2. Ta sẽ thực hiện hợp nhất từng cặp trong danh sách này. Kết quả sẽ là một node mới trong cây hợp nhất.

Xử lý đụng độ cập nhật / cập nhật

Trường hợp đụng độ tag name của element node sẽ được xử lý bằng ba option:

- Tự động chọn theo T_1
- Để người dùng tự chọn
- Chọn bằng cách đo khoảng cách ngữ nghĩa của tag name bằng cách sử dụng WordNet.

Trường hợp đụng độ text node cũng sẽ được xử lý bằng ba option:

- Tự động chọn theo T_1
- Để người dùng tự chọn
- Cho phép chỉnh sửa nội dung của Text node.

Để tăng tính hiệu quả cho công cụ, quá trình xử lý đụng độ cập nhật/cập nhật Text node sẽ được chia làm hai giai đoạn:

- Tính khác biệt giữa Text node thuộc T_B và T_1 và giữa T_B và T_2.
- Chọn lựa, chỉnh sửa.

Việc tính khác biệt nội dung của hai Text node được cho bởi thuật toán tính LCS [4].

3.4. Phát sinh tập khác biệt giữa hai cây và ráp cây

Để xuất ra tập khác biệt của hai tập tin XML T_1 và T_2, chúng ta sử dụng thuật toán tạo kịch bản chỉnh sửa [1] để có tập diff cực tiểu nhưng không tập trung vào việc diff có được mã hóa để thân thiện người dùng hay không, mục đích của chúng ta là tập diff có thể được ráp với T_2 để sinh ra T_1.

Để mô tả khác biệt giữa hai tập tin XML (hai cây có thứ tự), chúng ta sử dụng khái niệm kịch bản chỉnh sửa chi phí cực tiểu. Kịch bản chỉnh sửa chi phí cực tiểu của hai cây được định nghĩa bằng cách sử dụng các thao tác insert node, delele node, update node và move subtree làm các thao tác chỉnh sửa cơ bản. Ta chia thành hai bài toán con:

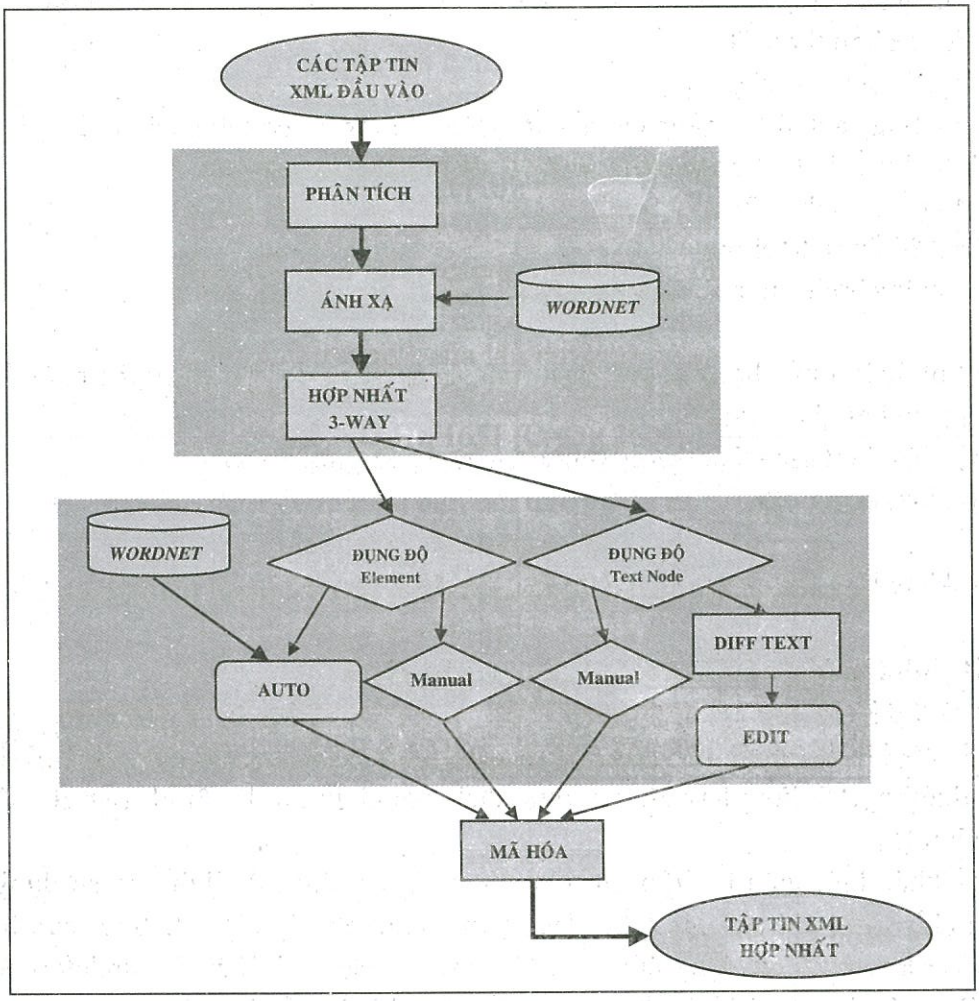
(1) tìm kiếm một ánh xạ giữa các đối tượng trong hai tập tin.

(2) tính kịch bản chỉnh sửa.

Nếu các đối tượng có các định danh duy nhất, bài toán thứ nhất được đơn giản hoá và chúng ta có thể sử dụng tính chất này để tăng tốc độ xử lý. Chúng ta sẽ sử dụng kết quả ánh xạ trong giai đoạn ánh xạ cây (chỉ sử dụng phần ánh xạ chính xác) để thực hiện tiếp việc tính kịch bản chỉnh sửa.

Việc tái phát sinh T_2 từ T_1 và kịch bản chỉnh sửa chỉ là áp dụng các thao tác được liệt kê trong kịch bản chỉnh sửa với T_1.

4. Kiến trúc tổng quát



Hình 4. Kiến trúc của công cụ hợp nhất Tập tin XML

Hình 4. trình bày kiến trúc tổng thể của chương trình. Công cụ đọc thông tin đầu vào từ hai hoặc ba tập tin XML và phân tích các tập tin thành các cấu trúc cây nội tại. Các node trong cấu trúc cây sau đó sẽ được ánh xạ với nhau.

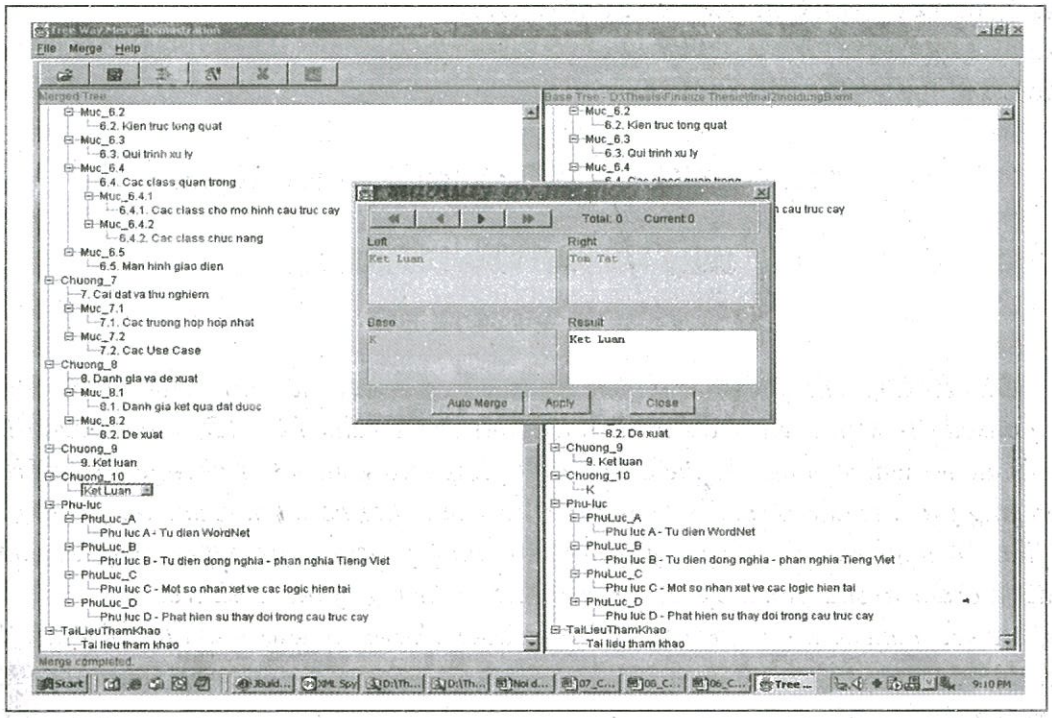
Điểm mạnh của kiến trúc này chính là cho phép tương tác với người dùng. Khi quá trình hợp nhất xảy ra các đụng độ, việc chọn lựa kết quả có thể được thực hiện tự động nhưng cũng có thể cho phép người dùng quyết định.

Công cụ được thiết kế trực quan và dễ sử dụng, thông tin đầu vào là các tập tin XML chuẩn.

Công cụ phát sinh tập khác biệt không yêu cầu một tri thức nào cộng thêm vì ở đây yêu cầu là tạo tập khác biệt cực tiểu và "ráp" lại cho đúng cây ban đầu chứ không cần phải đánh giá hoặc tìm kiếm một ánh xạ tốt theo nghĩa trong bài toán hợp nhất.

Màn hình giao diện

Màn hình kết quả hợp nhất



6. Kết luận

Bên cạnh các tinh chỉnh về ánh xạ mờ, trên cơ sở tiền đề đặt ra là việc hợp nhất chỉ dựa trên sự hỗ trợ tối thiểu, bài báo cũng đã cài đặt thử nghiệm hệ thống đo ngữ nghĩa bằng từ điển WordNet dựa trên CSDL prolog của Đại học Princeton nhưng đã chuyển đổi sang cơ chế thích hợp hơn để hỗ trợ quá trình đo ngữ nghĩa giữa hai node nhằm bước đầu xử lý quá trình ánh xạ và đụng độ một cách tinh tế và hữu ích hơn; do chúng ta đánh giá rằng tương lai, các máy di động có cài đặt sẵn WordNet là vấn đề rất thông dụng và nó chỉ chiếm một lượng không gian (cũng như thời gian xử lý) không đáng kể (khoảng < 5MB). Bên cạnh đó, để chứng minh vấn đề hỗ trợ đa ngôn ngữ của hệ thống, bài báo cũng đã cài đặt thử nghiệm Tự điển Đồng nghĩa-Phản nghĩa tiếng Việt (do hiện nay chưa có tự điển WordNet tiếng Việt). Để xử lý đụng độ bài báo phân biệt đụng độ TagName và đụng độ Text Node. Đối với đụng độ TagName, do tính gợi ý của các TagName, ta có thể sử dụng WordNet để chọn lựa tự động. Đối với đụng độ TextNode, thông qua thuật toán LCS, người dùng có thể nhận biết được các khác biệt giữa các nội dung đụng độ và có thể chọn lựa hoặc chỉnh sửa trực tiếp trên các TextNode bị đụng độ.

Chúng ta đã sử dụng việc mã hóa tập khác biệt dưới dạng một kịch bản chỉnh sửa cho phép cực tiểu hóa tập khác biệt, nhưng không sử dụng kịch bản này để hợp nhất. Với cách tiếp cận này, bài báo đã khắc phục được hai nhược điểm trong quá trình hợp nhất (tránh vấn đề đường dẫn cố định) và trong quá trình tạo khác biệt (cực tiểu hóa được tập khác biệt).

Các vấn đề cần được giải quyết tiếp theo để hoàn thiện công cụ bao gồm:

- Nghiên cứu cải tiến thuật toán hợp nhất 3-way, nhất là các trường hợp đụng độ cấu trúc.
- Nghiên cứu ứng dụng các thuật toán tạo khác biệt mới để có tập khác biệt càng nhỏ càng tốt.
- Thể hiện kịch bản chỉnh sửa cũng như các Tập tin XML dưới dạng thân thiện người dùng.

- Xử lý DTD (Document Type Definitions): hai tập tin có cấu trúc giống nhau nhưng DTD khác nhau cần phải được nhận biết.
- Xử lý dữ liệu dạng CDATA trong tập tin XML.
- Xử lý Tag Name có tên dài (phải xử lý ngôn ngữ).
- Xem xét ngữ nghĩa của Text Node.

MERGING STRUCTURED XMLFILES

Dong Thi Bich Thuy, Dinh Hung

***ABSTRACT**: The paper uses the approach of 3-way merging structured, take a XML file as a based file with entries are three XML files TB, T1 and T2 – TB is the based file,T1 and T2 are branches which conclude the diffrencies compared with TB or changes from TB.*
The merging process almost bases on the structure matches between branched files through the based file. To improve the accuracy of the matching process, the paper proposes an add on methode in order to sharpen it by using Wordnet to the tag of the XML tree.
Beside, the recommendation approach of this paper allows to detect and to solve conflicts that happen during merging process in some levels. The solution for solving these conflicts is the key of successful merging.

TÀI LIỆU THAM KHẢO

[1] *Chawathe S. S, Rajaraman A, Garcia-Molina H. and Widom J.* - **Change detection in hierarchically structured information**. - In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data.

[2] *George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller*– **Introduction to WordNet: An On-line Lexical Database** - (Revised August 1993)

[3] *IBM Alphaworks.* - **XML diff and merge tool home page**. http://www.alphaworks.ibm.com/tech/xmldiffmerge

[4] *J.W.Hunt* - **An Algorithm for Difference File Comparison** –Department of Electrical, *M. D. McIlroy,*Bell Laboratories, Murray Hill, New Jersey 07974 - Bell Laboratories Computing Science Technical, Report #41, dated July 1976.

[5] *Stefan Kurtz* - **Foundations of Sequence Analysis** - Lecture notes for a course in the Winter Semester 2000/2001 July 18, 2002

[6] **http://www.cogsci.princeton.edu/~wn/ - Five Papers on WordNet .**

[7] **http://www.deltaxml.com - DeltaXML Project**

[8] *Tancred Lindholm* - **A 3-way Merging Algorithm for Synchronizing Ordered Trees – the 3DM merging and differencing tool for XML** - Helsinki University of Technology, Dept. of Computer Science - September 13, 2001. http://www.cs.hut.fi/~ctl/3dm/