

GIẢI PHÁP PHẦN MỀM NGUỒN MỞ CHO WORKFLOW QUẢN LÝ HỒ SƠ CÔNG VĂN

Lâm Quang Vũ
Lê Trí Anh
Trần Hạnh Nhi
Võ Hữu Phúc
Đào Anh Vũ

TÓM TẮT

Trên thế giới hiện nay có nhiều giải pháp cho workflow quản lý nói chung, tuy nhiên các giải pháp này được công bố ở dạng thương mại và hạn chế sự can thiệp cũng như tùy chỉnh theo nhu cầu của đơn vị sử dụng. Trong đề tài này, chúng tôi sẽ nghiên cứu xây dựng một giải pháp thống nhất và hoàn chỉnh dựa trên các hệ thống nguồn mở rời rạc được xây dựng thô sơ sẵn có. Trước hết, để thiết kế quy trình phần mềm một cách trực quan thông qua giao diện đồ họa, chúng tôi thiết kế lại công cụ eclipse. Phần kết quả thiết kế được kết xuất qua một hệ thống xử lý chính workflow (core engine workflow) là jBPM chạy trên nền máy chủ JBoss. Ngoài ra, hệ thống chứng thực (Certificate authentication) dùng để cấp phát các chứng nhận số cho người dùng cũng như khóa chung, khóa riêng và chữ ký điện tử để người dùng có thể kiểm tra và ký trên các tài liệu điện tử. Để quản lý hệ thống người dùng và khai thác hệ thống chứng thực, chúng tôi xây dựng thêm phân quản lý người dùng tích hợp với hệ thống quản lý tài khoản LDAP. Ngoài ra, hệ thống quản lý tài liệu cũng được chúng tôi phát triển thêm kèm theo tính năng chuyển đổi hệ thống văn bản word sang dạng tập tin pdf có nhúng kèm chữ ký điện tử vào để người dùng có thể in ra.

1. Nhu cầu thực tế

Hiện tại việc cải cách hành chính đang được triển khai rộng khắp mọi ngành nghề trong xã hội, nhu cầu tin học hóa và chuẩn hóa hệ thống quản lý là một nhu cầu cấp thiết, việc triển khai các hệ thống luồng công việc (workflow) sẽ đem lại nhiều hiệu quả thiết thực trong việc quản lý của một tổ chức: giảm thời gian lưu chuyển hồ sơ (hồ sơ chuyển giao đến người nhận ngay lập tức), tăng tính tiện dụng (người dùng có thể theo dõi thông tin tình trạng hồ sơ mọi lúc, mọi nơi), tiết kiệm chi phí (chi phí giấy tờ, nhân sự phục vụ công tác hành chính), tiết kiệm nhân lực.

Các hệ thống nguồn mở hiện tại dùng để xây dựng hệ thống luồng công việc (workflow system) được phát triển và phát hành ở dạng rời rạc, riêng lẻ theo chuẩn của

tổ chức WfMC⁵ và OMG⁶ đưa ra. Chính vì lý do đó mà các hệ thống nguồn mở phải được xây dựng lại và phát triển một cách đồng bộ thống nhất từ phần thiết kế quy trình nghiệp vụ đến việc triển khai luồng công việc lên hệ thống để có thể triển khai thống nhất hệ thống luồng công việc nhằm đáp ứng nhu cầu thực tế.

2. Phân tích và đánh giá giải pháp

Điểm mới

So với các giải pháp nguồn mở khác như wfmOpen, XFlow, Taverna,

⁵ WfMC – Workflow Management Coalition – <http://www.wfmc.org>

⁶ OMG – The Object Management Group – <http://www.omg.org/>

EnhydraShark, ...⁷, jBPM cung cấp các module cơ bản ở mức tổng quát hóa cao nhất để có thể phát triển thành một hệ thống workflow hoàn chỉnh. Điểm mới so với các giải pháp khác chính là sự linh động và tính mở của các module do jBPM phát triển. Điều này cho phép nhà phát triển xây dựng thêm và tùy biến lại cho phù hợp với ngữ cảnh sử dụng.

Điểm mới khác của giải pháp jBPM được thể hiện ở chỗ các thành phần lõi của jBPM được xây dựng trên nền Java thuần túy là J2SE để quản lý việc định nghĩa quy trình và các thể hiện của quy trình. Chính vì được xây dựng trên nền J2SE nên jBPM có thể được đóng gói thành dạng EJB để triển khai trên các máy chủ hỗ trợ J2EE.

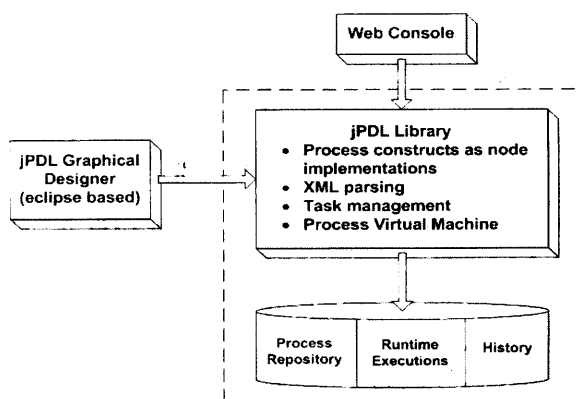
Qua quá trình khảo sát các hệ thống workflow nguồn mở, hệ thống jBPM được công bố rộng rãi trong cộng đồng có bao gồm mã nguồn của hệ thống lõi jBPM. Điểm đặc biệt trong việc công bố mã nguồn này là hệ thống tài liệu đặc tả sơ lược và cộng đồng hỗ trợ jBPM được xây dựng tương đối hoàn chỉnh.

Tuy vậy, mã nguồn của hệ thống jBPM được phát hành ở các phiên bản đã lỗi thời và có tiềm ẩn những lỗi chưa được kiểm tra và công bố. Cộng đồng hỗ trợ chủ yếu là những tổ chức hoặc cá nhân đã có kinh nghiệm và đã sử dụng qua hệ thống jBPM.

Với các giải pháp workflow nguồn mở khác như wfmOpen, XFlow, Taverna, EnhydraShark, ...³ kiến trúc của các hệ thống này không linh động và mềm dẻo như kiến trúc có thể cắm thêm được của jBPM. Chính vì vậy, hệ thống jBPM có thể được phát triển và triển khai thêm theo nhu cầu thực tế sử dụng của tổ chức. Tính linh động cao hơn hẳn so với các hệ thống workflow nguồn mở khác.

Qua quá trình khảo sát các lõi hệ thống luồng công việc với yêu cầu đáp ứng được các tiêu chí trên, hệ thống jBPM được tích

hợp chạy trên máy chủ JBoss của hãng JBoss đã được lựa chọn.



Hình 2. Hệ thống jBPM ban đầu

Hệ thống jBPM ban đầu được công bố ở mức thô sơ, chỉ cung cấp các ký hiệu cơ bản để biểu diễn luồng công việc. Ngoài ra, hệ thống eclipse dùng để thiết kế luồng công việc (editor) cũng rất hạn chế trong việc hỗ trợ thiết kế đầy đủ các trường hợp như tạo nhiều thể hiện (multi-instance). Việc tích hợp các thành phần riêng rẽ lại với nhau để tạo thành một hệ quản lý luồng công việc hoàn chỉnh với chi phí thấp và khả năng tùy biến cao nhất cũng là một yếu tố trở ngại cho việc đưa hệ thống luồng công việc vào thực tế. Hơn nữa, việc xây dựng các hệ thống chứng thực tài liệu, chữ ký điện tử trên tài liệu và các chứng thực kèm theo trong luồng công việc cũng chưa được triển khai.

Để triển khai hệ thống luồng công việc hoàn chỉnh trên nền tảng ứng dụng jBPM vào thực tế, cần phải xây dựng thêm các thành phần hỗ trợ khác như:

Cải tiến trình soạn thảo quy trình nghiệp vụ (business process) jPDL Editor.

Cải tiến phần lõi thực thi jBPM để có thể xử lý tạo nhiều thể hiện (multi-instant).

Xây dựng phần quản lý tài liệu lưu chuyên trong luồng công việc (Document manage system).

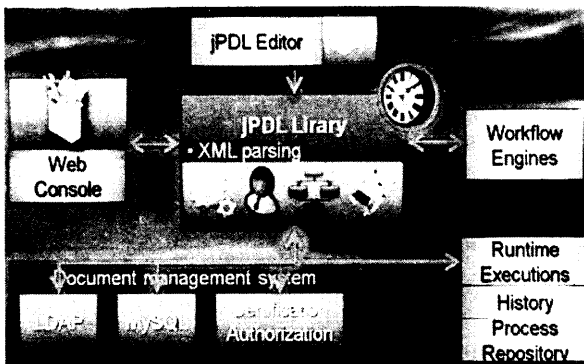
⁷ Tài liệu tham khảo.

Tăng cường tính tiện dụng và các tiện ích quản lý trên giao diện của hệ thống jBPM.

Xây dựng phần tích hợp với hệ thống LDAP.

Cấu hình jBPM sử dụng với cơ sở dữ liệu MySQL.

Xây dựng và tích hợp thêm phần chứng thực tài liệu, chứng thực người dùng và các vấn đề bảo mật với hệ thống jBPM.



Hình 3. Hệ thống jBPM sau khi cài tiến

Điểm hạn chế

Các module được cung cấp còn thiếu nhiều chức năng cơ bản để có thể triển khai thành một giải pháp hoàn chỉnh: trình biên tập quy trình nghiệp vụ (chưa xây dựng thêm ký hiệu (notation), phần xử lý, phần lịch biểu và bộ lập lịch), hệ thống lõi jbpm (phần xử lý document, phân quản lý người dùng, phân phân quyền, phân ánh xạ tài liệu, ... cũng chưa được xây dựng)

Phiên bản miễn phí của máy chủ JBoss được cung cấp để ứng dụng thực thi vận hành chưa ổn định (hay bị lỗi tràn bộ nhớ, lỗi không nhận được sự thay đổi phía client, sau 24h kể từ khi start server – server không nhận tín hiệu phản hồi từ client) (giải pháp tạm thời là tăng bộ nhớ ảo để tránh phát sinh lỗi tràn bộ nhớ).

Về mặt ngôn ngữ dùng để đặc tả quy trình nghiệp vụ, hệ thống jBPM sử dụng ngôn ngữ jPDL. Ưu điểm của ngôn ngữ này là tính mở rộng cao, cho phép người thiết kế có thể bổ sung xây dựng thêm các khái niệm

mới để biểu diễn quy trình cụ thể. Tuy nhiên, ngôn ngữ jPDL không thuộc dòng ngôn ngữ chính thống do các tổ chức như WfMC hay OMG đưa ra. Chính vì vậy mà vấn đề chuẩn hóa các khái niệm biểu diễn quy trình nghiệp vụ cũng như các ký hiệu chưa mang tính chuẩn mực như các ngôn ngữ XPDL, BPEL hay WS-BPEL, ...

So với các hệ thống workflow nguồn mở khác như XFlow, wfmOpen, Taverna, EnhydraShark, ...³ hệ thống jBPM còn một số điểm hạn chế như về mặt chuẩn hóa ngôn ngữ mô hình hóa quy trình nghiệp vụ được sử dụng. Tuy nhiên, những ưu điểm và điểm mới của jBPM như khả năng cắm thêm được, mã nguồn được công bố đầy đủ và hệ thống tài liệu hỗ trợ tương đối hoàn chỉnh so với các hệ thống khác đã đáp ứng được nhu cầu tùy biến hóa một hệ thống workflow nguồn mở hoàn chỉnh để áp dụng vào thực tế.

3. Khó khăn trong quá trình xây dựng và giải pháp

Hệ thống nội tại của jBPM chỉ hỗ trợ các chức năng đơn giản như quản lý quy trình không có kèm theo document. Chính vì nhu cầu quản lý document, ban đầu nhóm đã thực hiện khảo sát OpenKM, đây là phần mềm nguồn mở hỗ trợ việc quản lý CSDL truyền đi trong hệ thống. Tuy nhiên, phần mềm này chỉ mở version thấp nhất, dù mã nguồn được công bố, nhưng tài liệu hỗ trợ kèm theo không có và không có cộng đồng nguồn mở hỗ trợ, nên nhóm đã quyết định chuyển sang phân tích và xây dựng một hệ thống quản lý tài liệu riêng rẽ.

Mặc dù jBPM có cung cấp bộ thiết kế quy trình đồ họa GPD, tuy nhiên mã nguồn của GPD phiên bản mới nhất không được công bố và mã nguồn của phiên bản cũ hơn lại không được kiểm chứng, phiên bản đóng gói công bố trên trang sourceforge không gặp vấn đề lỗi, nhưng mã nguồn công bố lại mắc phải một số lỗi khi thực thi.

Riêng phần editor hoàn toàn không có tài liệu đặc tả kèm theo. Ban đầu khi lựa chọn hệ thống hỗ trợ xây dựng quy trình

ng nghiệp vụ với giao diện đồ họa. Nhóm đã lựa chọn NexusBPM, đây là hệ thống hỗ trợ thiết kế quy trình nghiệp vụ đồ họa với tài liệu đặc tả và hỗ trợ đáng tin cậy. NexusBPM bao gồm cả phần mã nguồn dạng đầy đủ (phát triển dựa trên eclipse và hoàn toàn được tạo ra chỉ để thiết kế quy trình nghiệp vụ đồ họa), ngoài ra còn có dạng plugins vào hệ thống eclipse sẵn có.

jBPM cung cấp mã nguồn của GPD, tuy nhiên mã nguồn này lại không có bất kỳ tài liệu hướng dẫn nào đi kèm. Chính vì vậy mà nhóm phải dựa trên tìm hiểu NexusBPM để có thể thiết kế thêm và chỉnh sửa mã nguồn của jBPM GPD.

Khả năng mở rộng là một ưu thế mà ngôn ngữ đặc tả quy trình jPDL chiếm được so với hệ thống các ngôn ngữ khác. Tuy nhiên, vấn đề đặt ra là khi xây dựng thêm document node, hệ thống file định nghĩa quy trình đã không còn ổn định nữa, nghĩa là khi này nếu không xây dựng phần triển khai phía server ở một hệ thống mới, thì hệ thống mới sẽ không thể hiểu và xử lý được document node được thêm vào.

Với những ngôn ngữ được qui định nghiêm ngặt và xây dựng theo tiêu chuẩn của các tổ chức workflow thì khả năng chỉnh sửa và cải tiến sẽ vô cùng khó khăn, tuy nhiên, quy trình thiết kế với những ngôn ngữ này sẽ có thể thực thi trên mọi server hỗ trợ ngôn ngữ đó mà không cần phải quan tâm đến việc bổ sung phần xử lý cho server.

Giải pháp cho vấn đề đa thể hiện (multi-instance): khi một tác vụ được phân công cho một nhóm người. Khi thực thi quy trình, tác vụ được phân công cho nhiều người sẽ được tạo ra thành nhiều phiên bản giống nhau để từng người trong nhóm có thể hoàn tất tác vụ của mình. Hệ thống ban đầu chưa xây dựng phần đặc tả đa thể hiện một tác vụ khi thiết kế quy trình lần phần thực thi đa tác vụ, với nhu cầu thể hiện đa tác vụ nên phần đa thể hiện các tác vụ đã được xây dựng thêm. Do đó, muốn phân công cùng một tác vụ cho nhiều người, người thiết kế chỉ cần thiết lập cấu hình cho tác vụ đó ngay trong

quá trình tạo quy trình nghiệp vụ và sau đó triển khai qua hệ thống máy chủ jBPM. Việc cấu hình số lượng thể hiện của tác vụ sẽ dựa vào nhóm (group), bằng cách phân công tác vụ đó cho một nhóm, khi thực thi quy trình, danh sách các thành viên nhóm sẽ được lấy ra và tác vụ sẽ được gán lần lượt cho từng thành viên của nhóm thông qua lớp thư viện được xây dựng thêm trong phần lõi của jBPM.

Giải pháp cho vấn đề tái sử dụng: khi thiết kế và triển khai một quy trình nghiệp vụ, khả năng tái sử dụng lại quy trình ở một đơn vị khác cũng là vấn đề được đặt ra trong quá trình phát triển hệ thống luồng công việc. Các mức độ tái sử dụng được phân chia như sau:

Tái sử dụng quy trình đã được định nghĩa: khả năng tái sử dụng quy trình được định nghĩa có nghĩa là khi quy trình nghiệp vụ được thiết kế và triển khai một lần, với một quy trình tương tự nhưng chỉ khác thành phần tham gia, người dùng có thể bắt đầu một quy trình mới nhanh chóng nhất mà không phải xây dựng lại từ đầu. Giải pháp cho vấn đề này chính là cho phép tổng quát hóa các thành phần tham gia vào quy trình khi xây dựng. Với mỗi quy trình được vận hành, thao tác chỉ định cụ thể các thành phần tham gia sẽ giúp người dùng có thể tái sử dụng lại quy trình một cách hoàn chỉnh nhất. Trong đó có hai thành phần chính là tái sử dụng tài liệu được định nghĩa và tái sử dụng người tham gia vào quy trình.

Tái sử dụng tài liệu đã được định nghĩa: với tên trừu tượng được sử dụng trong quá trình thiết kế quy trình. Khi triển khai, thao tác ánh xạ tài liệu thực và tài liệu trừu tượng sẽ được thực hiện.

Tái sử dụng người tham gia đã được định nghĩa: tương tự như giải pháp với tài liệu, người tham gia trong quá trình thiết kế luồng công việc cũng được chỉ định bằng cách cung cấp tên ở mức tổng quát (như chức danh, hoặc vai trò). Khi triển khai, thao tác ánh xạ cũng lại được thực hiện để cụ thể hóa người tham gia vào quy trình.

Chính vì khả năng tổng quát hóa người tham gia và tài liệu khi thiết kế quy trình nghiệp vụ và khả năng ánh xạ các khái niệm tổng quát với các khái niệm cụ thể nên luồng công việc có khả năng tái sử dụng cao mà không cần phải xây dựng lại từ đầu.

Các tài liệu tham gia vào quy trình sẽ được ký bởi người tham gia bằng chữ ký điện tử. Với những tài liệu cần phải có sự xác nhận và chấp thuận, chữ ký điện tử của người tham gia quy trình sẽ được dùng để ký nhằm đảm bảo tính xác thực và giá trị của tài liệu.

4. Kết luận và hướng phát triển

Xây dựng hệ thống hỗ trợ tạo quy trình dạng ad-hoc

Trong thực tế, có những quy trình sẽ không được định nghĩa trước, ví dụ như việc chuyển thông báo. Khi người A nhận được thông báo và chuyển cho người B, người B có thể tự do chuyển thông báo mà không cần chỉ định trước, quy trình này kết thúc ở một bước bất kỳ không xác định trước. Đây là quy trình dạng ad-hoc.

Hiện nay, hệ thống jBPM hoàn toàn không hỗ trợ tính năng này, cho nên một hướng mở rộng khác của jBPM đó là xây dựng thêm tính năng hỗ trợ quản lý ad-hoc, dạng bất kỳ để phục vụ nhu cầu của người sử dụng hệ thống.

Đây là một vấn đề thực tế có tính cần thiết cao, tuy nhiên cần phải có thời gian để nghiên cứu chi tiết phương án giải quyết bài toán này.

Nghiên cứu vấn đề triển khai quy trình lúc runtime

Trong thực tế, một quy trình có thể không được định nghĩa một cách toàn diện và đầy đủ, người dùng có thể tạo một bước X trong quy trình mà bước X này là kết quả thực hiện của một quy trình khác mà quy trình này không được chỉ định trước lúc thiết kế quy trình. Đó là dạng triển khai sub process lúc runtime.

Hiện tại, hệ thống jBPM đã hỗ trợ triển khai quy trình với một bước bất kỳ có thể là kết quả thực hiện của quy trình khác. Tuy nhiên, quy trình này phải được thiết kế và triển khai đồng bộ lúc xây dựng quy trình nghiệp vụ.

Về mặt nguyên lý hoạt động, hệ thống jBPM sẽ triển khai đồng thời quy trình chính và mọi quy trình con (nếu có). Khi thông tin kích hoạt gửi đến node có quy trình con, thì quy trình con sẽ được gọi thực hiện, khi này quy trình chính sẽ ở trạng thái chờ cho đến khi hoàn tất quy trình con. Do đó, để thực hiện quy trình bất kỳ, khi này cần can thiệp sâu vào hệ thống jBPM để xây dựng cho người dùng quyền chỉ định thực thi quy trình con sẵn có hoặc dạng ad-hoc.

Tài liệu tham khảo

1. Vondrak: "Business process modeling and workflow automation".
2. Cesare Pautasso: "Lecture 11- Workflow Management Systems", Computer Science Management, Swiss Federal Institute of Technology (ETHZ).
3. Tom Baeyens: "The State of Workflow", May-2004, tom@jbpm.org
4. JBoss jBPM - "Workflow in Java", <http://docs.jboss.org/jbpm/v3.2/-userguide/html/>
5. The Object Management Group (OMGT) <http://www.omg.org/>
6. The Business Process Management Initiative <http://www.bpmi.org/>
7. jBPM business process management, user guide <http://www.jboss.org/jboss-jbpm/jpdl/>
8. State of art of standards in business process modeling and execution: iisl.postech.ac.kr/publication/thesis/2007_jjm.pdf, <http://iisl.postech.ac.kr/>
9. T.Stoilov, K.Stoilova : "E-business workflow modeling and execution

- tools”, Department Hierarchical Systems, Institute of Computer and Communication Bulgarian Academy of Sciences, Acad. G.Bonchev str. B1.2, 1113 Sofia, Bulgaria.
10. Domain Specific Languages, <http://www.martinfowler.com/bliki/Domain-SpecificLanguage.html>.
 11. Workflow Pattern, <http://www.workflowpatterns.com/>.
 12. Process virtual machine , manual page, http://docs.jboss.org/jbpm/pvm/manual/html_single/.
 13. Introduction of the Graphical process designer, <http://www.jboss.org/jbossjbpm/gpd/>
 14. Eclipse plugin, step by step, power by IBM, <http://www.ibm.com/developerworks/opensource/library/os-ecplug>
 15. Plugin development environment <http://www.eclipse.org/articles/Article-PDE-does-plugins/PDE-intro.html>
 16. Eclipse Ganymede help, <http://help.eclipse.org/ganymede/index.jsp>
 17. An introduction of Eclipse architecture and technical problems, http://www.eclipse.org/articles/Article-Plug-in-architecture/plugin_architecture.html
 18. Understanding how Eclipse plug-ins work with OSGi, <http://www.ibm.com/developerworks/opensource/library/os-ecl-osgi/>
 19. Moving Resources Between Eclipse and Rational Software Architect, http://open.ncsu.edu/se/tutorials/import_export
 20. A First Look at Eclipse Plug-In Programming , http://www.developer.com/java/other/article.php/10936_3316241_2
 21. Plug-in Development Environment (PDE) in Eclipse, http://open.ncsu.edu/se/tutorials/plugin_dev/
 22. Eclipse and Jazz, <http://openseminar.org/se/modules/18/index/screen.do>
 23. Open Source Product Evaluation <http://www.workflowpatterns.com/evaluations/opensource/index.php>
 24. Workflow patterns, introductions and evaluations, <http://www.workflowpatterns.com/documentation/index.php>
 25. wfmOpen, <http://www.wfmopen.com>
 26. XFlow, <http://www.xflow.com>
 27. EnhydraShark, <http://www.enhydrashark.com>
Taverna, <http://www.taverna.com>