



[123doc] - tai-lieu-su-dung-framework-activiti-va-bpmn

Cấu trúc dữ liệu và giải thuật (Trường Đại học Nguyễn Tất Thành)



Scan to open on Studocu

MỤC LỤC

A. BPMN.....	4
I. Định nghĩa.....	4
II. Phạm vi.....	4
III. Các loại phần tử cơ bản trong BPMN 2.0.....	5
1. <i>Flow objects</i>	5
2. <i>Connecting objects</i>	5
3. <i>Swim lanes</i>	5
4. <i>Artifacts</i>	5
IV. Flow objects và connecting objects.....	5
• Flow objects.....	5
1. <i>Event</i>	5
➤ Start event.....	6
➤ Intermediate event.....	6
➤ End event.....	6
2. <i>Activity</i>	6
➤ Task.....	6
➤ Sub-process.....	6
➤ Transaction.....	7
➤ Call Activity.....	7
3. <i>Gateway</i>	7
➤ Exclusive.....	7
➤ Event Based.....	7
➤ Parallel.....	8
➤ Inclusive.....	8
➤ Exclusive Event Based.....	8
➤ Complex.....	8
➤ Parallel Event Based.....	8

•	Connections.....	8
1.	<i>Sequence Flow</i>	8
2.	<i>Message Flow</i>	8
3.	<i>Association</i>	8
V.	Swimlanes và artifacts.....	9
•	Swimlanes.....	9
1.	<i>Pool</i>	9
2.	<i>Lane</i>	9
•	Artifacts.....	9
B.	Framework Activiti.....	10
I.	Các thành phần cơ bản của Framework Activiti.....	10
1.	<i>ProcessEngineConfiguration</i>	13
2.	<i>ProcessEngine</i>	13
3.	<i>RepositoryService</i>	13
4.	<i>TaskService</i>	14
5.	<i>IdentityService</i>	14
6.	<i>FormService</i>	14
7.	<i>RuntimeService</i>	15
8.	<i>ManagementService</i>	15
9.	<i>HistoryService</i>	15
II.	Database.....	15
III.	Hoạt động của luồng công việc.....	17
IV.	Cách sử dụng một số thành phần cơ bản trong biểu đồ luồng công việc.....	19
1.	<i>StartEvent</i>	19
2.	<i>ServiceTask</i>	19
3.	<i>UserTask</i>	19
4.	<i>ScriptTask</i>	19
5.	<i>Sub-process</i>	20

6.	<i>MailTask</i>	20
7.	<i>EndEvent</i>	20

Framework Activiti & BPMN

A. BPMN

I. Định nghĩa

- Business Process Model and Notation (BPMN) là chuẩn để mô hình hóa các tiến trình nghiệp vụ bằng cách cung cấp các kí hiệu đồ họa dùng để đặc tả các tiến trình nghiệp vụ trong biểu đồ tiến trình nghiệp vụ (BPD – Business Process Diagram), dựa trên một luồng biểu đồ kỹ thuật giống với biểu đồ hoạt động trong ngôn ngữ mô hình hóa thống nhất (UML- Unified Modeling Language). Mục tiêu của BPMN là hỗ trợ quản lý các tiến trình nghiệp vụ, cho cả người dùng kỹ thuật và người dùng doanh nghiệp, bằng cách cung cấp các ký hiệu trực quan cho người dùng doanh nghiệp, chưa có khả năng thể hiện các tiến trình có ngữ nghĩa phức tạp. Đặc điểm kỹ thuật của BPMN cũng cung cấp một tập các ánh xạ giữa kí hiệu đồ họa và ngôn ngữ thực thi của kiến trúc bên dưới, cụ thể là ngôn ngữ thực thi tiến trình nghiệp vụ (BPEL – Business Process Execution Language).
- Mục đích cơ bản của BPMN là để cung cấp một chuẩn ký hiệu có thể đọc hiểu bởi tất cả các nghiệp vụ các bên liên quan. Chúng bao gồm các nhà phân tích nghiệp vụ, người tạo và cải tiến các tiến trình, các nhà phát triển kỹ thuật chịu trách nhiệm phát triển chúng và các nhà quản lý kinh doanh, người giám sát và quản lý chúng. Do đó, BPMN phục vụ như là một ngôn ngữ chung, thu hẹp khoảng cách thường xuyên xảy ra giữa thiết kế và thực hiện tiến trình kinh doanh.

II. Phạm vi

- BPMN là ràng buộc chỉ để hỗ trợ cho các khái niệm về mô hình áp dụng cho các quy trình quản trị kinh doanh. Các loại mô hình được thực hiện bởi các tổ chức không vì mục đích xử lý là nằm ngoài phạm vi của BPMN. Ví dụ về mô hình nằm ngoài phạm vi của BPMN là:
 - + Cơ cấu tổ chức (Organizational structures)
 - + Phân tích chức năng (Functional breakdown)
 - + Mô hình dữ liệu (Data models)

- Ngoài ra, BPMN còn cho thấy luồng của dữ liệu (thông điệp) và kết hợp với ***, nó không phải là một biểu đồ luồng dữ liệu.

III. Các loại phần tử cơ bản trong BPMN 2.0

- Mô hình BPMN bao gồm các biểu đồ đơn giản được xây dựng từ một tập các ràng buộc của các phần tử đồ họa. Đối với cả người dùng doanh nghiệp và các nhà phát triển, họ đơn giản hóa việc hiểu các luồng và tiến trình hoạt động kinh doanh. Bốn loại phần tử cơ bản của BPMN là:

1. *Flow objects*

- *Events, activities, gateways*

2. *Connecting objects*

- Sequence flow, message flow, association

3. *Swim lanes*

- Pool, lane

4. *Artifacts*

- Data object, group, annotation
- Bốn loại này cho phép tạo các sơ đồ tiến trình nghiệp vụ đơn giản (BPDs). BPDs cũng cho phép làm mới các kiểu của flow object hoặc artifact, để làm cho biểu đồ dễ hiểu hơn.

IV. Flow objects và connecting objects

✚ *Flow objects*

- Flow objects là thành phần mô tả chính trong BPMN, và gồm có 3 phần tử cốt lõi: events, activities và gateways.

1. *Event*

- Một sự kiện (Event) được biểu diễn bằng một hình tròn và biểu thị một cái gì đó xảy ra so sánh với một hoạt động (activity), đó là một cái gì đó được thực hiện). Các biểu tượng trong các vòng tròn biểu thị kiểu của sự kiện (ví dụ, một phong bì thể hiện đại diện cho một thông điệp hoặc một cái đồng hồ đại diện cho thời gian). Các sự kiện cũng được phân loại thành Catching (ví dụ, bắt một thông báo đến bắt đầu

một tiến trình) hoặc Throwing (chẳng hạn ném một thông điệp hoàn thành khi kết thúc một tiến trình).

➤ Start event

- Các hoạt động giống như một tiến trình kích hoạt (process trigger); được thể hiện bằng một đường viền đơn nhỏ và chỉ có thể được bắt (Catch), vì vậy nó được thể hiện bằng biểu tượng mở (phác thảo).

➤ Intermediate event

- Thể hiện những điều xảy ra giữa bắt đầu và kết thúc các sự kiện; được biểu diễn bằng đường viền kép và có thể được ném (Throw) hoặc bắt (Catch) (sử dụng biểu tượng cố định (solid) hoặc mở (open) thích hợp). Ví dụ, một tác vụ có thể theo luồng đề****

➤ End event

- Thể hiện kết quả của một tiến trình; được biểu diễn bởi một đường viền đơn dày hoặc đậm và chỉ có thể ném (Throw), vì vậy nó được biểu diễn bằng một biểu tượng cố định.

2. Activity

- Một hoạt động được thể hiện bằng một hình chữ nhật góc tròn và mô tả loại công việc cần phải được thực hiện.

➤ Task

- Một tác vụ đại diện cho một đơn vị công việc mà nó không hoặc không thể chia nhỏ tới một cấp độ chi tiết cao hơn của tiến trình nghiệp vụ mà không có biểu đồ các bước thực hiện trong thủ tục (đó không phải là mục đích của BPMN).

➤ Sub-process

- Được sử dụng để che giấu hoặc trình bày các mức thêm chi tiết cho tiến trình nghiệp vụ. Khi thu gọn, một tiến trình con được thể hiện bởi một dấu cộng so với dòng dưới cùng của hình chữ nhật; khi mở rộng, các hình chữ

nhật tròn mở rộng để hiển thị tất cả các đối tượng luồng, các đối tượng kết nối và các vật (artifacts).

- Nó có các sự kiện bắt đầu và kết thúc của riêng mình; các luồng tuần tự từ tiến trình cha không phải vượt qua đường biên.

➤ Transaction

- Một hình thức của tiến trình con trong đó tất cả các hoạt động bên trong phải được coi như là một thể thống nhất; ví dụ, tất cả phải được hoàn thành để đáp ứng một mục tiêu, và nếu có bất kỳ một trong số chúng thất bại thì chúng phải được đền bù (hoàn tác). Các phiên giao dịch được phân biệt với các tiến trình con mở rộng được bao quanh bởi một đường viền kép.

➤ Call Activity

- Một điểm trong một tiến trình nơi mà một tiến trình chung hoặc một tác vụ chung được tái sử dụng. Một call activity được phân biệt với các kiểu activity khác bởi một đường viền đậm bao quanh vùng hoạt động.

3. Gateway

- Một cổng được biểu diễn với một hình viên kim cương và được xác định sự phân nhánh và các đường dẫn tùy thuộc vào điều kiện thể hiện.

➤ Exclusive

- Được sử dụng để tạo các luồng thay thế trong một tiến trình. Bởi vì chỉ có một trong những đường có thể được thực hiện nên nó được gọi là độc quyền (exclusive).

➤ Event Based

- Điều kiện để xác định đường của một tiến trình dựa trên một sự kiện được đánh giá.

➤ Parallel

- Được sử dụng để tạo đường song song mà không đánh giá bất kỳ điều kiện nào.

➤ Inclusive

- Được sử dụng để tạo luồng thay thế nơi mà tất cả các đường đều được đánh giá.

➤ Exclusive Event Based

- Một sự kiện đang được đánh giá để xác định các đường dẫn loại trừ lẫn nhau sẽ được thực hiện.

➤ Complex

- Được sử dụng để mô hình hóa các hành vi đồng bộ hóa phức tạp.

➤ Parallel Event Based

- Hai tiến trình song song được bắt đầu dựa trên một sự kiện, nhưng không đánh giá sự kiện này.

✚ **Connections**

- Flow objects được kết nối với nhau sử dụng Connecting objects, nó có ba loại là: sequences, messages and associations.

1. Sequence Flow

- Một luồng tuần tự được biểu diễn với một đường nét liền và mũi tên ở đầu và thể hiện****

2. Message Flow

- Một luồng thông điệp được biểu diễn bằng một đường nét đứt, một hình tròn mở là điểm bắt đầu và một mũi tên mở là điểm kết thúc.

3. Association

- Một Association được biểu diễn bằng một đường chấm chấm. Nó được sử dụng để kết hợp một vật (Artifact) hoặc một văn bản tới một đối tượng luồng (Flow Object) và có thể chỉ ra một số hướng sử dụng mũi tên mở (hướng về phía vật thể để biểu diễn kết quả, hướng từ vật ra thể hiện đầu vào, đọc và cập nhật). Không có hướng được sử dụng

khi vật thể hoặc văn bản có liên quan tới một luồng tuần tự hoặc luồng thông điệp (chẳng hạn như luồng đã cho thấy rõ hướng).

V. Swimlanes và artifacts

Swimlanes

- Swim lanes là một cơ chế trực quan hóa tổ chức và phân loại các hoạt động, dựa trên biểu đồ luồng chức năng chéo (cross functional flowcharting) và trong BPMN gồm có 2 loại:

1. *Pool*

- Đại diện cho những thành phần chính tham gia trong một tiến trình, đặc biệt

2. *Lane*

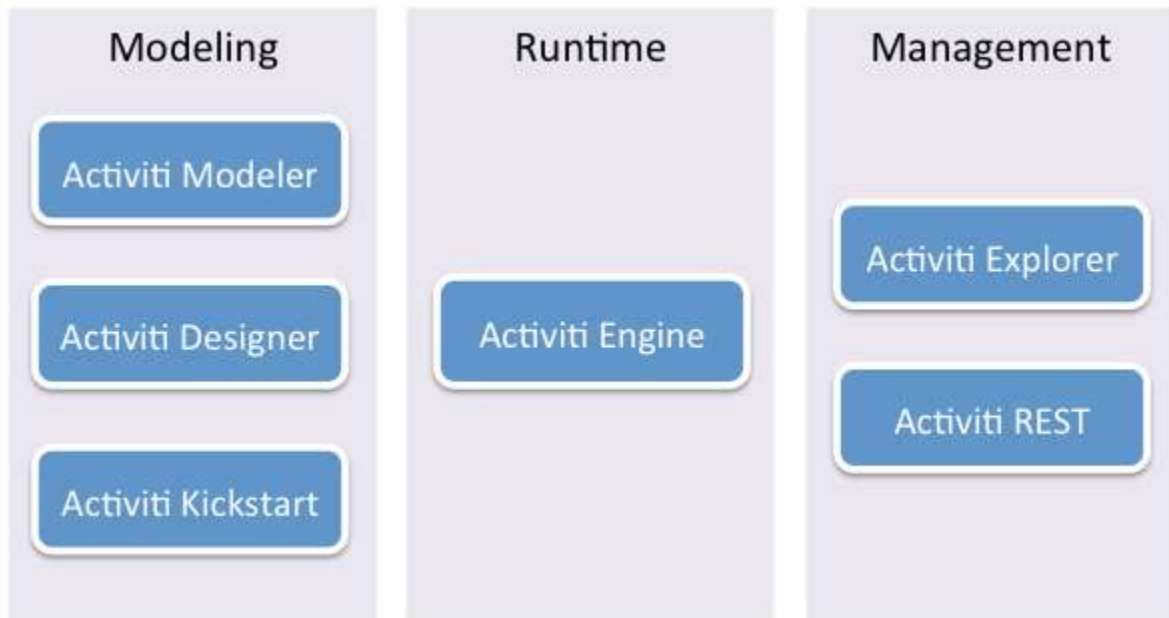
-

Artifacts

- Cho phép những người phát triển có thể chứa một vài thông tin bên trong mô hình/biểu đồ. Bằng cách này thì mô hình/biểu đồ sẽ dễ đọc hơn. Có ba Artifacts được định nghĩa sẵn và chúng là:
 - + Data objects: Đối tượng dữ liệu thể hiện người đọc
 - + Group:
 - + Annotation: Một thông báo được sử dụng để đưa ra

B. Framework Activiti

I. Các thành phần cơ bản của Framework Activiti



Hình

- Có 3 thành phần cơ bản là Modeling, Runtime và Management.
- + Modeling: Gồm 3 thành phần cơ bản
 - Activiti Modeler:
 - ✓ Activiti Modeler có thể từng được sử dụng để xác thực
 - Activiti Designer:
 - ✓
 - Activiti Kickstart:
 - ✓
- + Runtime:
 - Activiti Engine:
 - ✓ Đặc điểm: Đây là thành phần quan trọng của Activiti, nó được coi là trái tim của Activiti.

- Cho phép người dùng cập nhật có thể kết hợp với các tiến trình cập nhật giao dịch đơn.
- Có thể chạy trên bất kỳ môi trường Java nào như Spring, JTA, độc lập với bất kỳ hình thức phân chia giao dịch nào.
- Dễ dàng gọi đến và chạy các các tiện ích được cài đặt.
- Thêm các hoạt động một cách đơn giản và

✓

+ Management:

- Activiti Explorer: Là một ứng dụng web ví dụ cung cấp khả năng truy cập đến Activiti Engine runtime cho tất cả người dùng của hệ thống. Nó gồm có quản lý tác vụ (task management), kiểm duyệt tiến trình cụ thể (process instance inspection), quản lý đặc trưng và hiển thị báo cáo (management features and viewing reports) trên cơ sở các dữ liệu thống kê cũ.

✓ Task management:

- Các trường hợp sử dụng chính:
 - + Hiển thị danh sách tác vụ của các cá nhân.
 - + Hiển thị danh sách tác vụ mà người dùng hiện tại là 1 ứng viên.
 - + Tạo một tác vụ mới (không liên quan đến tiến trình).
 - + Hoàn thành một tác vụ bằng cách gửi dữ liệu qua task form.
 - + Phân công lại một tác vụ cho người dùng khác.
 - + Xem những người dùng khác có tham gia tiến trình liên quan đến tác vụ này.
 - + Tạo tác vụ con và phân công chúng cho mọi người.
- Các trường hợp quản lý sử dụng:
 - + Hiển thị danh sách các tác vụ của thuộc cấp.

- Các trường hợp chủ sở hữu tiến trình sử dụng:
+ Kiểm duyệt chi tiết tiến trình cụ thể liên quan đến một tác vụ.

✓ Management:

- Thể hiện nếu Activiti Engine bật và đang chạy hoặc vấn đề xảy ra nằm ở đâu?
- Quản lý các bản triển khai.
- Hiển thị tài nguyên của triển khai.
- Quản lý các định nghĩa tiến trình.
- Kiểm duyệt các bảng của cơ sở dữ liệu.
- Hiển thị các activiti log.
- Thời gian trung bình để thực hiện một phiên giao dịch.
- Công việc đó đã thất bại bao nhiêu lần? Nguyên nhân là gì? Nhóm việc công việc với các lỗi tương ứng. Thử lại công việc.

✓ Reports

- Tất cả các sự kiện liên quan đến quá trình thực thi tiến trình cụ thể sẽ được lưu trữ lại trong một cơ sở dữ liệu lịch sử tiến trình. Từ các bước/hành động trong những định nghĩa tiến trình ban đầu là

✓ Process instance inspection

-

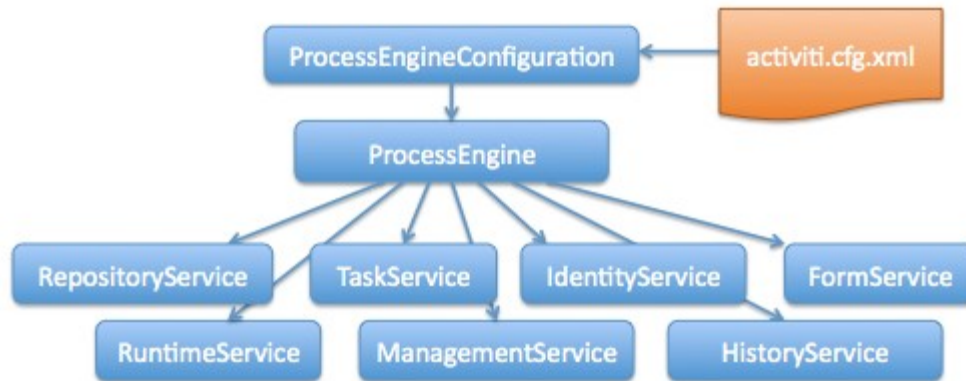
✓ Table based process design

-

➤ Activiti REST:



- Các thành phần của Activiti Engine là:



Hình . Các thành phần cơ bản của Activiti Engine.

1. *ProcessEngineConfiguration*

- Dùng để cấu hình ProcessEngine từ file activiti.cfg.xml.

2. *ProcessEngine*

- ProcessEngine sẽ được xây dựng theo cách của Spring:
 - + Đầu tiên, Spring application context sẽ được tạo ra.
 - + Sau đó, ProcessEngine được tạo ra từ application context.

3. *RepositoryService*

- Là service đầu tiên cần sử dụng khi làm với Activiti engine.
- Service này cung cấp các toán tử cho quản lý và thao tác triển khai và định nghĩa process.
- RepositoryService cho phép triển khai như những package.
- Service này cho phép:
 - + Truy vấn trong quá trình triển khai và định nghĩa process.
 - + Đình chỉ và kích hoạt toàn bộ tiến trình hoặc định nghĩa một tiến trình cụ thể.
 - + Lấy tài nguyên khác nhau như các tập tin chứa trong các sơ đồ triển khai hoặc quá trình tự động được tạo ra bởi các công cụ.
 - + Lấy một phiên bản của tiến trình được định nghĩa mà có thể được sử dụng để quan sát quá trình sử dụng Java hơn XML.

4. TaskService

- TaskService cần được sử dụng bởi người dùng của hệ thống. Nó là thành phần quan trọng của BPM Engine cũng như của Activiti. Mọi thứ liên quan đến tác vụ đều được nhóm lại trong TaskService như:
 - + Truy vấn các tác vụ được giao cho người dùng hoặc nhóm.
 - + Tạo mới các tác vụ độc lập. Đây là những tác vụ không liên quan đến một tiến trình cụ thể nào.
 - + Thao tác để tạo tác vụ giao cho người dùng.
 - + Claiming và completing một tác vụ. Claiming có nghĩa là một người nào đó thông báo

5. IdentityService

- IdentityService khá đơn giản.
- Cho phép quản lý (tạo, cập nhật, xóa, truy vấn ...) của các nhóm và người sử dụng.
- IdentityService sẽ không thực hiện bất kỳ xác thực trên người dùng trong thời gian chạy. Activiti có thể làm điều đó thông qua việc kết hợp các cơ chế, dịch vụ như LDAP, kích hoạt thư mục

6. FormService

- Các FormService là service tùy chọn. Tức là Activiti có thể sử dụng không có nó mà không bị mất bất kỳ chức năng nào.
- Dịch vụ này đưa ra các khái niệm về start form và task form:
 - + Một start form là một form được hiển thị cho người dùng trước khi tiến trình cụ thể được bắt đầu.
 - + Một task form là form được hiển thị khi người dùng muốn hoàn tất một form.
- Activiti cho phép xác định các form trong quá trình định nghĩa BPMN 2.0.
- Dịch vụ này cho phép các dữ liệu có thể làm việc với nhau một cách dễ dàng. Nhưng nó không thực sự cần thiết.

7. *RuntimeService*

- Được dùng để chạy tiến trình đã được cài đặt và thực thi các câu lệnh truy vấn đến các tiến trình trong quá trình thực thi luồng công việc.

8. *ManagementService*

- Các ManagementService thường không cần thiết khi mã hóa ứng dụng tuy chỉnh sử dụng Activiti.
- ManagementService cho phép lấy các thông tin trong bảng cơ sở dữ liệu và bảng siêu dữ liệu.
- Nó có khả năng truy vấn và quản lý các toán tử của jobs.
- Jobs được sử dụng trong Activiti một cách đa dạng như thời gian, tiếp diễn không đồng bộ, trễ đình chỉ/kích hoạt....

9. *HistoryService*

- Các HistoryService cho ta thấy tất cả các dữ liệu cũ được thu thập bởi công cụ Activiti.
- Khi thực hiện tiến trình, rất nhiều dữ liệu có thể được lưu giữ bởi các công cụ (ở đây có khả năng cấu hình) như thời gian bắt đầu của một tiến trình cụ thể, những người dùng thực hiện tác vụ, thời gian để hoàn thành tác vụ
- Service này chủ yếu có khả năng truy vấn để truy cập cơ sở dữ liệu.

II. **Database**

- ACT_RE_*: 'RE' là viết tắt của repository. Những bảng này sẽ lưu trữ các thông tin có chứa tiền tố "static" như các định nghĩa tiến trình (process definitions) và tài nguyên tiến trình (process resources) (như ảnh, rules...).

Gồm có:

+ ACT_RE_MODEL

+ ACT_RE_DEPLOYMENT

+ ACT_RE_PROCDEF

- ACT_RU_*: 'RU' là viết tắt của runtime. Những bảng này sẽ lưu trữ các thông tin khi thực hiện, bao gồm các dữ liệu của tiến trình cụ thể (process

instances), user task, biến, công việc... Activiti chỉ lưu trữ các dữ liệu chạy trong quá trình thực thi tiến trình cụ thể và loại bỏ các bản ghi khi một tiến trình cụ thể kết thúc. Những lưu trữ trong bảng thực thi rất nhỏ và nhanh chóng.

Gồm có:

+ ACT_RU_EVENT_SUBSCR

+ ACT_RU_IDENTITYLINK

+ ACT_RU_JOB

+ ACT_RU_TASK

+ ACT_RU_VARIABLE

+ ACT_RU_EXECUTION

- ACT_ID_*: 'ID' là viết tắt của từ identity. Những bảng chứa các thông tin định danh như người sử dụng, các nhóm ...

Gồm có:

+ ACT_ID_INFO

+ ACT_ID_MEMBERSHIP

+ ACT_ID_USER

+ ACT_ID_GROUP

- ACT_HI_*: 'HI' là viết tắt của từ history. Những bảng này có chứa các dữ liệu lịch sử như là tiến trình cụ thể cũ, task cũ, biến cũ

Gồm có:

+ ACT_HI_ACTINST

+ ACT_HI_ATTACHMENT

+ ACT_HI_COMMENT

+ ACT_HI_DETAIL

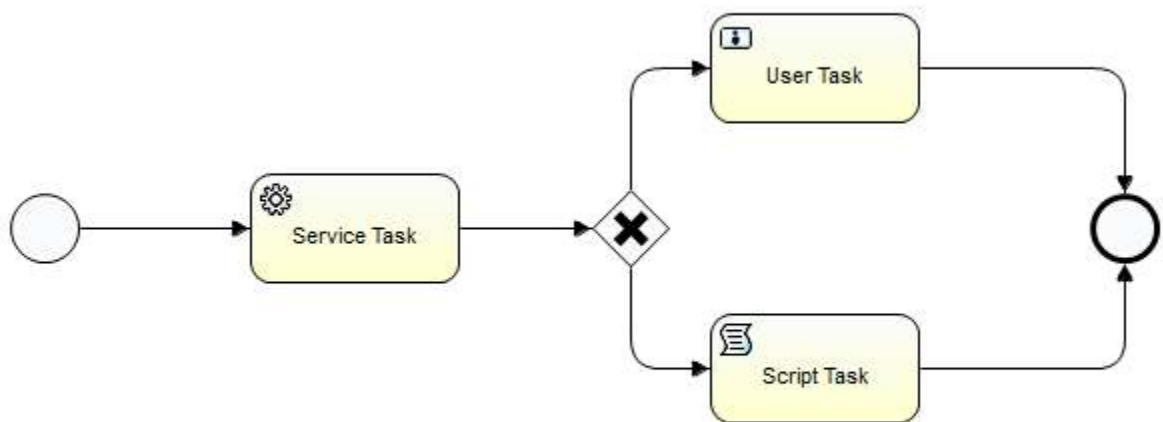
+ ACT_HI_IDENTITYLINK

+ ACT_HI_PROCINST

- + ACT_HI_TASKINST
 - + ACT_HI_VARINST
 - ACT_GE_*: 'GE' là viết tắt của từ general. Bảng này lưu trữ các dữ liệu chung, được sử dụng trong nhiều trường hợp khác nhau.
- Gồm có:
- + ACT_GE_PROPERTY
 - + ACT_GE_BYTEARRAY
 - Ngoài ra, còn có thêm bảng ACT_EVT_LOG cùng với sequence ACT_EVT_LOG_SEQ dùng để lưu trữ log của các sự kiện xảy ra trong quá trình thực thi các luồng công việc.

III. Hoạt động của luồng công việc

- Ví dụ một luồng công việc như sau:



Hình . Biểu đồ luồng đặc tả bằng BPMN 2.0

- Các thành phần bên trong gồm có:
 - + Start Event: Bắt đầu luồng xử lý không điều kiện.

- + ServiceTask: Các xử lý dữ liệu đầu vào hoặc cung cấp dữ liệu đầu vào để tiến hành th
- + UserTask: Các xử lý liên quan đến tài khoản người dùng, nhóm người dùng...
- + ScriptTask: Các kịch bản xử lý được định nghĩa sẵn bằng một ngôn ngữ script nhất định.
- + ExclusiveGateway: Cổng định tuyến, dùng để rẽ nhánh luồng xử lý dựa theo kết quả được trả lại từ ServiceTask.
- + EndEvent: Kết thúc một luồng công việc.
- Các bước thực hiện luồng công việc trên:
 - + Bước 1: Tạo kết nối với cơ sở dữ liệu và tiến hành định nghĩa các Engine mặc định của Activiti.
 - + Bước 2: Khởi tạo biểu đồ tiến trình từ file xml hoặc bpmn.
 - + Bước 3: Tiến hành định nghĩa các tác vụ và ghi vào các bảng thực thi trong cơ sở dữ liệu.
 - + Bước 4: Tiến hành thực hiện các tác vụ đã được định nghĩa.
 - + Bước 5: Ghi lại lịch sử và xóa dữ liệu tại các bảng thực thi.

IV. Cách sử dụng một số thành phần cơ bản trong biểu đồ luồng công việc

1. StartEvent

- Là điểm bắt đầu của một tiến trình, tiến trình con, tác vụ.
- Có nhiều kiểu StartEvent với các tham số đầu vào khác nhau như Error, Message....

2. ServiceTask

- Trong ServiceTask có hỗ trợ 3 kiểu Task là:
 - + Java class: Cần có 1 file java được implement từ org.activiti.engine.delegate.JavaDelegate để tiến hành các xử lý bên trong.
 - + Expression: Cần một biểu thức logic.

+ Delegate Expression: Cần một biểu thức dạng Delegate.

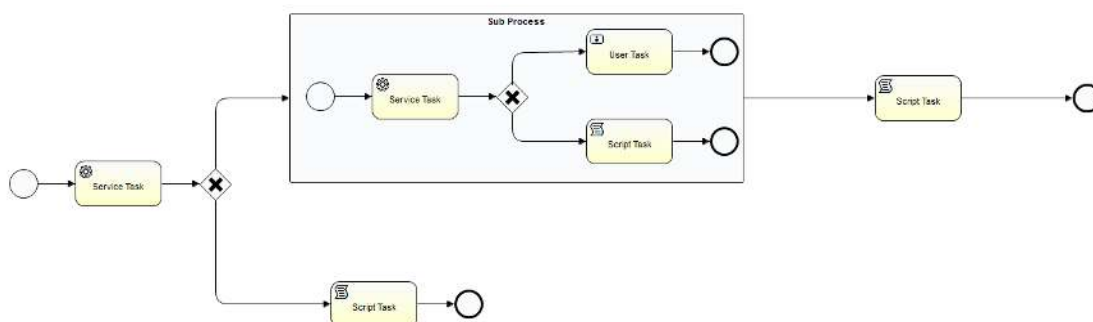
3. *UserTask*

- Dùng để xử lý các thao tác liên quan đến xác thực quyền sử dụng của người dùng.
- Sử dụng các tham số định danh nhóm, quyền, mức độ ưu tiên.

4. *ScriptTask*

- Dùng để chạy các kịch bản đã được thể hiện thông qua ngôn ngữ script.
- Trong Activiti hỗ trợ 2 kiểu ngôn ngữ script là javascript và groovy.

5. *Sub-process*



Hình-

- Chứa các tiến trình con được định nghĩa bên trong luồng công việc chính.
- Tiến trình con cũng bao gồm các thành phần tương tự như các tiến trình thông thường, nhưng nó được thiết lập các sự kiện dành riêng cho nó (Event Sub-process).
- Thường được kết sử dụng để kết nối các tiến trình có liên quan với nhau thành một luồng công việc chung để xử lý các dữ liệu riêng biệt mà không ảnh hưởng đến logic của luồng chính.
- Trong luồng chính, luồng con được coi như một task thông thường nên có thể liên kết và thực hiện tuần tự hoặc kết hợp với các task khác trong luồng công việc.

6. MailTask

- Đảm nhận vai trò nhận và gửi email trong quá trình thực thi luồng công việc.
- Được chia ra làm 2 loại riêng biệt là: Received Task và Send Task tương ứng với từng chức năng riêng biệt là nhận và gửi email.

7. EndEvent

- Đánh dấu một điểm kết thúc của một tiến trình hoặc tiến trình con.
- Thường được gọi đến khi kết thúc luồng công việc hoặc xảy ra các sự kiện lỗi trong quá trình thực thi tiến trình.