|  |
| --- |
| Macintosh HD:Users:jh7rnn:Dropbox:Citeck:EcoS:Дизайн 2015:Citeckecos.png  On Alfresco ECM Platform |
| **Citeck EcoS Case Management mapping to CMMN 1.1** |

|  |
| --- |
| Author: Simonov Pavel  [pavel.simonov@citeck.ru](mailto:pavel.simonov@citeck.ru) |
| 2016 |

**Table of contents**

[Charts of entities 3](#_Toc451756775)

[Сiteck EcoS 3](#_Toc451756776)

[Case Management Model and Notation 4](#_Toc451756777)

[Reworking 5](#_Toc451756778)

****Charts of entities****

****Сiteck EcoS****

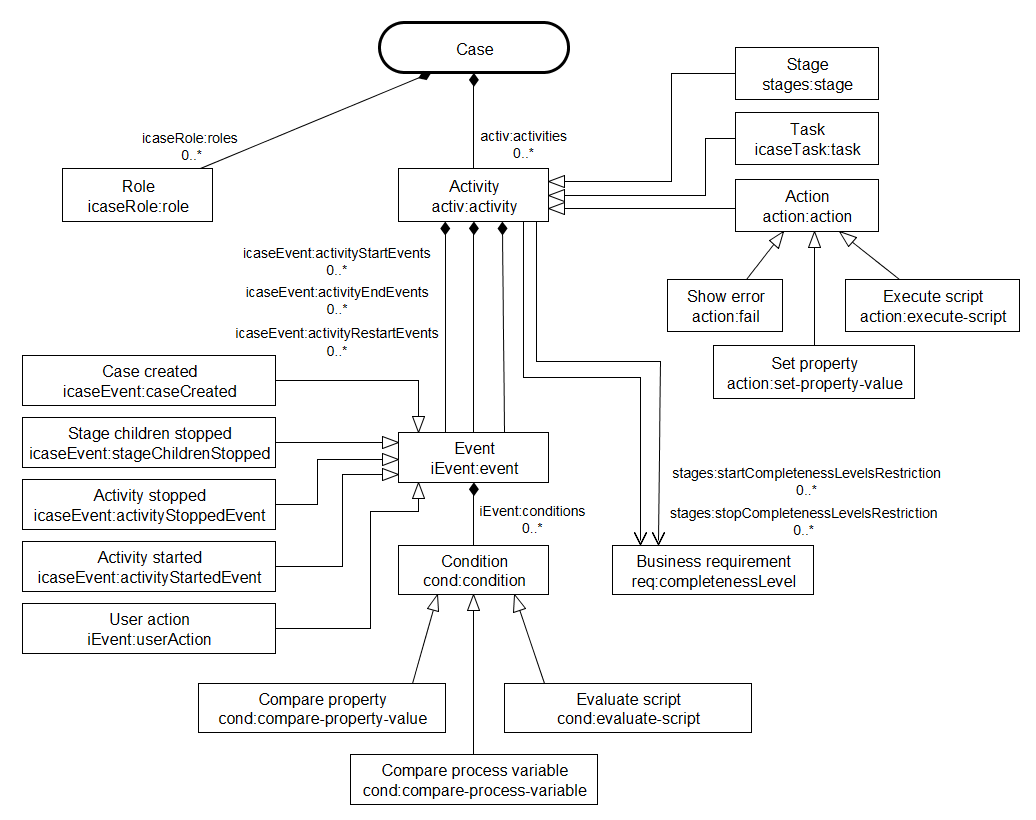


Figure1 – Entities’ chart of case management in CiteckEcoS

|  |  |
| --- | --- |
| * Case * Role * Activity * Stage * Task * Action * Show error * Set property * Execute script * Business requirement | * Event * Case created * Stage children stopped * Activity stopped * Activity started * User action * Condition * Compare property * Evaluate script * Compare process variable |

Case Management Model and Notation

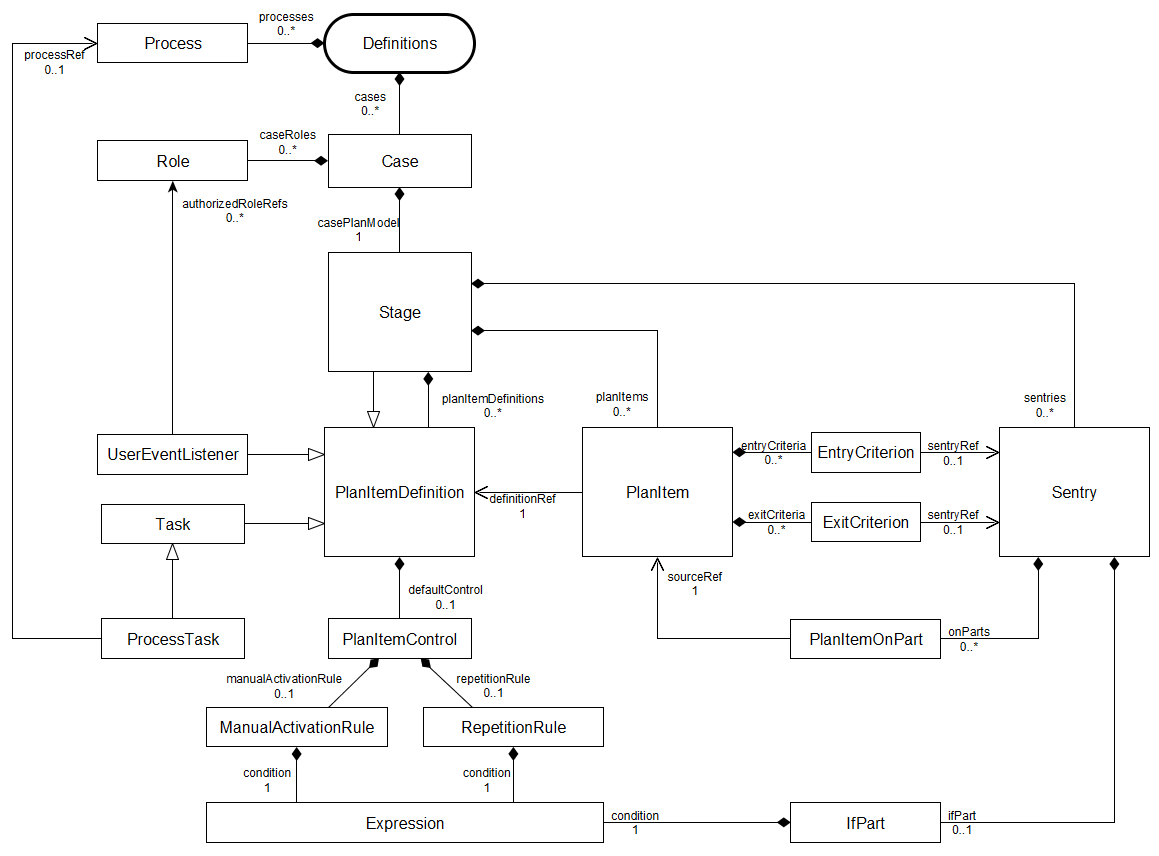


Figure 2 – Entities’ chart of case management in CMMN 1.1\*

|  |  |
| --- | --- |
| * Definitions * Case * Role * PlanItemDefinition * Stage * Task * ProcessTask * UserEventListener * PlanItem * Process | * Criterion * EntryCriterion * ExitCriterion * Sentry * PlanItemOnPart * IfPart * Expression * PlanItemControl * ManualActivationRule * RepetitionRule |

\* illustrated only those entities which functional is implemented in CiteckEcoS

Reworking

1. AuthorizedRoles

a) add the *icaseRole:hasAuthorizedRoles* aspect to the roles’ model *(icaseRoleModel.xml)* with one multiple association *icaseRole:authorizedRoles*;

b) add *icaseRole:authorizedRoles* field to the *iEvent:userAction* creation form;

c) Complement the *CaseActionsProvider* with a verification code of a current user presence in the *authorizedRoles* list (directly in one of roles or within one of group selected to the role). If there is a user, an action will be restored, if not, the action will be hidden.

1. Flag ‘repeatable’in activity

a) add the *activ:repeatable* boolean property to the (icaseActivityModel.xml ->activ:activity)activities model;

b) add a new property to the activities creation forms (*stage, task, action*);

c) expand the “startActivity” method in CaseActivityServiceImpl by adding there logic for activities restart:

When the *startActivity* method was called: If an activity is completed, one checks a “repeatable” flag and if it is *true, resetActivity* methodis called and then code from *startActivity* should be executed

d) complete an invariant for relevance for *icaseEvent:activityRestartEvents.* This field should be hidden if it is empty and shown only if it is filled in;

e) Comment above the icaseEvent:hasActivityRestartEvents aspect on the fact it is out of date and there is *repeatable* flag instead of it;

f) Create a script to transfer all events from an *icaseEvent:activityRestartEvents* association to *icaseEvent:activityStartEvents* and a *repeatable* flag installation if there were such associations.

1. Stage stages:autoComplete

a) add the stages:autoComplete flag to the (stagesModel.xml ->stages:stage) stages model;

b) add a flag to creation/editing forms;

c) remove the icaseEvent:stageChildrenStopped creation form;

d) In the CaseActivityEventTrigger a method tryToFireStageChildrenStoppedEvents:

- Change a method name to *tryToCompleteStages;*

- Replace *eventService.fireEvent* to *caseActivityService.stopActivity;*

- Add verification for a presence of the *autoComplete* flag (only if it is *true,* call *caseActivityService.stopActivity*)