Tempo
**INTALIO**Persistence

Log In

Added by Jackie Ju, last edited by Ark Xu on Dec 07, 2008

# Background

## Purpose of this document

This document is to focus on the persistence layer form Tempo and Humtan task, compare their current behavior. With the difference, we can check what needs to be extends for Tempo to comply with Human Tash specification.
The description of this topic is draft, any comments or suggestions are appreciated.

## Terminology

HT: Human Task

# Task meta data:

For Tempo, the Task meta data is located in pipa file, the deployment service will scan all the pipa files under the configure directory, parse these files and perpist them with PIPATask record.
But on the PIPA file, there only seven properties have been configured.

| Property | Comments |
|---|---|
| task-role-owners | For owners |
| task-user-owners | For users |
| userProcessInitSOAPAction | ?Is it to specify the who will initialize the task? |
| processEndpoint | Endpoint |
| formNamespace | xForm namespace |
| formURI | URL for the xForm file |
| task-description | task description |

These seven properties are far away from the HT task requirement. This document will instroduce what the data needs to be persistanced at design time, focus on the below points:

- Task Interface definition.
- People Management(It is about how to assign one user to one role and to execute one action)
- PresentationElements.

## Task Interface definition.

This element is used to specify the operation used to invoke the task.
Properties in HT:
portType, operation, responsePortType, responseOperation.
Properties in Tempo:
processEndpoint, userProcessInitSOAPAction.

The property processEndpoint is one concrete webservice address, it looks like: "http://localhost:8080/ode/processes/AbsenceRequest/Employee_To_AbsenceRequest". But the HT interface definition doesn't speicify the concrete address information. Also don't know what is the meaning of property userProcessInitSOAPAction in Tempo.
Need further check for this part. Any comments?

## User Mangement.

The essential relationship is the user and role. In Tempo, it is straightforward, below class diagram comes from Tempo.
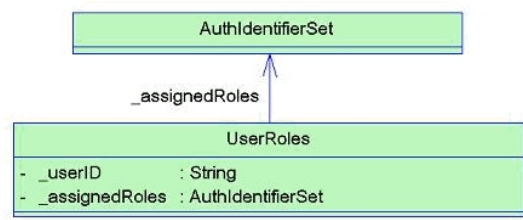


In Human Task definition, the user management is more flexiable. It support to query the user with specific qarameter, then assign the queryed user to logic group, originational entity or roles.
The user can be assigned with three differnt ways:
1): Via Logic people groups.
2): Via literals.
3): Via expression.

Check below sampels:
1): Manage the user with logic people groups, the below sample to query the user for the "Shanghai" regin.

```
<htd:logicalPeopleGroups>
    <htd:logicalPeopleGroup name="regionalClerks">
        <htd:parameter name="region" type="xsd:string" />
    </htd:logicalPeopleGroup>
</htd:logicalPeopleGroups>

  <htd:peopleAssignments>
    <htd:potentialOwners>
      <htd:from logicalPeopleGroup="regionalClerks">
          <htd:argument name="region">
              Shanghai
          </htd:argument>
      </htd:from>
    </htd:potentialOwners>
```

2): Manage user with literals:

```
<htd:potentialOwners>
```

```
        <htd:from>
            <htd:literal>
                <htd:organizationalEntity>
                    <htd:users>
                        <htd:user>Alan</htd:user>
                        <htd:user>Dieter</htd:user>
                    </htd:organizationalEntity>
            </htd:literal>
        </htd:from>
    </htd:potentialOwners>
```

3): Manage user with expressions:

```
    <htd:potentialOwners>
        <htd:from>
            htd:getInput("part1")/approvers
        </htd:from>
    </htd:potentialOwners>
```

Exclude to the second case(assign people via literal), all the other assignments support to assign people with runtime query by some specific parameters. And Tempo only support the second case currently.

For the ACL:
In Tempo, the authentication is called ACL(Access Control List?). The ACL is going to check which user/role are authenticated to execute one specific action with one specific task.
ACL class diagram:
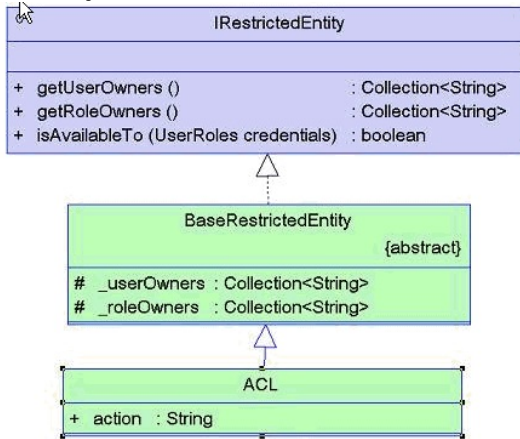


**IRestrictedEntity**: It represents one entity needs the access control.

In HT, assign one user to one role through the tag <htd:from>, the role type has descided who can execute one action or not. The logic is totally differntly different for this ACL part.
Check the <tFrom> and <tGenericHumanRole> definition:

```
    <xsd:complexType name="tFrom" mixed="true">
      <xsd:complexContent>
        <xsd:extension base="tExtensibleMixedContentElements">
          <xsd:sequence>
            <xsd:choice>
              <xsd:element name="argument" type="tArgument" minOccurs="0" />
              <xsd:element name="literal" type="tLiteral" minOccurs="0" />
            </xsd:choice>
          </xsd:sequence>
          <xsd:attribute name="expressionLanguage"  type="xsd:anyURI" />
          <xsd:attribute name="logicalPeopleGroup" type="xsd:QName" />
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="tGenericHumanRole">
      <xsd:complexContent>
        <xsd:extension base="tExtensibleElements">
          <xsd:sequence>
            <xsd:element name="from" type="tFrom" />
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
```
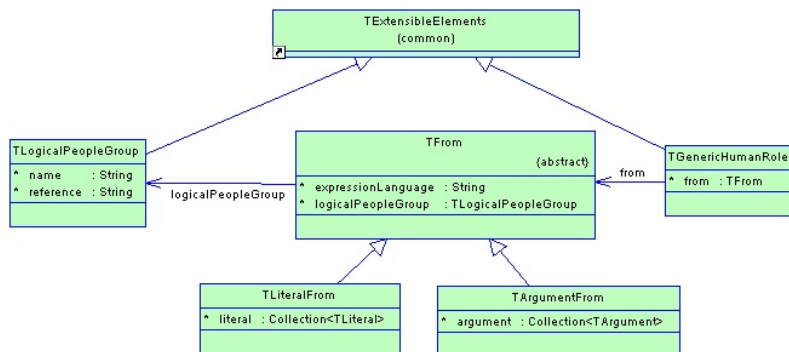
Additionally, it needs to default support six generic Human roles in HT task meta data.
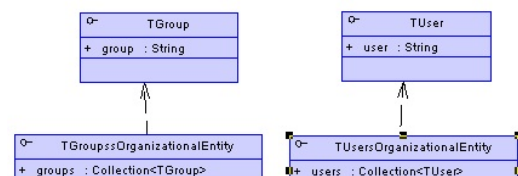They are Task initiator, Task stakeholders, Potential owners, Excluded owners, Business administrators, Notification recipients.

Below is the draft class diagram based on HT spec, Tempo may needs to align with it.
Generic Human Role diagram:



Organizational Entities diagram:

## PresentationElements.

It is to define what is the data will be showed on the page, the data may comes from the Task instance, or just some text.
Corresponding presentation element in Tempo is the xForm, with the rough check, I think the change is minimal to cover Human Task funtion.

Below is one example for the PresentationElements for Human Task:

```
<htd:presentationElements>
        <htd:name xml:lang="en-US">Approve Claim</htd:name>
        <htd:name xml:lang="de-DE"> Genehmigung der Schadensforderung </htd:name>
        <htd:presentationParameters>
          <htd:presentationParameter name="firstname" type="xsd:string">
            htd:getInput("ClaimApprovalRequest")/cust/firstname
          </htd:presentationParameter>
          <htd:presentationParameter name="lastname" type="xsd:string">
            htd:getInput("ClaimApprovalRequest")/cust/lastname
          </htd:presentationParameter>
          <htd:presentationParameter name="euroAmount" type="xsd:double">
            htd:getInput("ClaimApprovalRequest")/amount
          </htd:presentationParameter>
        </htd:presentationParameters>
        <htd:subject xml:lang="en-US">
          Approve the insurance claim for $euroAmount$ on behalf of $firstname$ $lastname$
        </htd:subject>
        <htd:subject xml:lang="de-DE">
          Genehmigung der Schadensforderung $euroAmount$ $firstname$ $lastname$
        </htd:subject>
        <htd:description xml:lang="en-US" contentType="text/plain">
          Approve this claim following corporate guideline #4711.0815/7 ...
        </htd:description>
        <htd:description xml:lang="en-US"
          contentType="text/html">
          <p>Approve this claim following corporate guideline <b>#4711.0815/7</b></p>
        </htd:description>
        <htd:description xml:lang="de-DE" contentType="text/plain">
          Genehmigen Sie diese Schadensforderung entsprechend Richtlinie Nr. 4711.0815/7 ...
        </htd:description>
        <htd:description xml:lang="de-DE"
          contentType="text/html">
          <p> Genehmigen Sie diese Schadensforderung entsprechend Richtlinie <b>Nr. 4711.0815/7</b>
          </p>
        </htd:description>
      </htd:presentationElements>
```

## Property Comparation for Task meta data:

PIPATask class diagram:
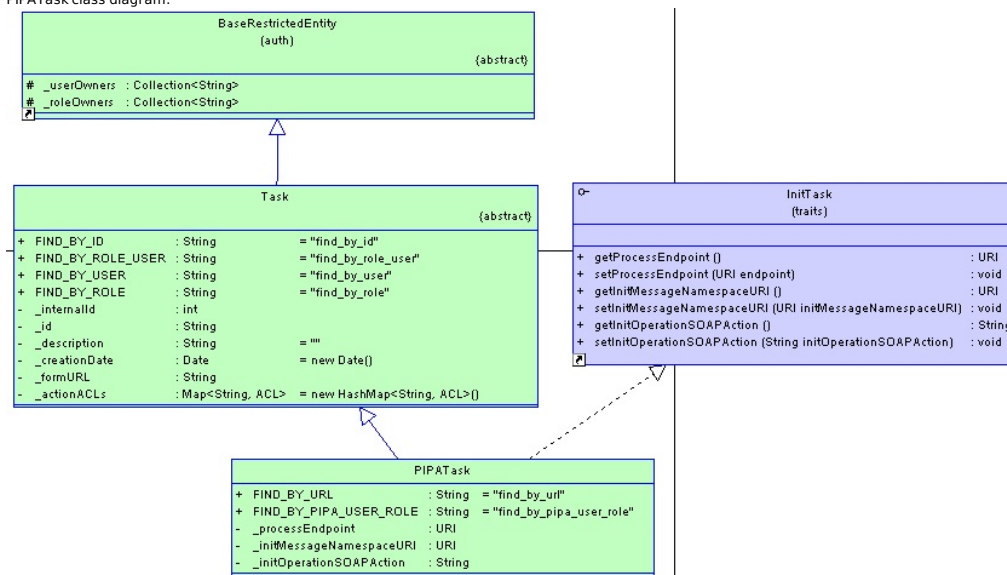


Compare the properties between Tempo PIPATask with HT task scehma definition.

| PIPATask | Property in Human Task | Comments |
|---|---|---|
| processEndpoint | interface | It is to specify the operation used to invoke the task in HT, check the section: Task Interface definition |
| N/A | priority | non negative integer. HT: supports expression evaluation. Tempo: Hasn't mentioned in PIPA file |
| actionACLs | tPeopleAssignments data type | Check the section: People management section. Tempo: Hasn't mentioned in PIPA file |
| N/A? | tDelegation data type | HT: support four values: anybody, potentialOwners, other, nobody. |
| formURL | tPresentationElements data type | HT: specify different information used to display the task in a task list. Tempo: use the xForm for the representation |
| N/A | output(tQuery data type) | HT: It is to indentity the field of output message which reflects the result. Tempo: Hasn't mentioned in PIPA file |
| N/A | searchBy(tExpression data type) | HT: it is to search for task instance based on one customer search criterion. Tempo: Hasn't mentioned in PIPA file |
| formURL? | tRenderings | HT: is used to specify the rendering method. Tempo: Need further investigate on this part. |
| N/A | deadlines(tDeadlines data type) | HT: for the deadline information. Tempo: Hasn't mentioned in PIPA file. |
| description? | name(Qualified name) | HT: it is one required attribute. |
| internalId | N/A | Tempo: is it used internally? |
| id | N/A | Tempo: is it used internally? |
| creationDate | N/A | HT: the creation date is part of instance data. |

## Task Instance data:

In Tempo the PATask is much closer to HT task instance definition, check the PATask class diagram.
PATask diagram:

**IChainableTask**
(traits)

| | |
|---|---|
| + isChainedBefore () | : boolean |
| + setChainedBefore (boolean flag) | : void |
| + getPreviousTaskID () | : String |
| + setPreviousTaskID (String id) | : void |

**IProcessBoundTask**
(traits)

| | |
|---|---|
| + getProcessID () | : String |
| + setProcessID (String processID) | : void |

**ITaskWithDeadline**
(traits)

| | |
|---|---|
| + getDeadline () | : Date |
| + setDeadline (Date deadline) | : void |

**PATask**

| | | |
|---|---|---|
| + FIND_BY_STATES | : String | = "find_by_ps_states" |
| + FIND_BY_PA_USER_ROLE | : String | = "find_by_pa_user_role" |
| + FIND_BY_PA_USER_ROLE_GENERIC | : String | = "find_by_pa_user_role_gener |
| - _processID | : String | |
| - _state | : TaskState | = TaskState.READY |
| - _failureCode | : String | = "" |
| - _failureReason | : String | = "" |
| - _completeSOAPAction | : String | |
| - _input | : String | |
| - _output | : String | |
| - _attachments | : Map<String, Attachment> | = new HashMap<String, Attach |
| - _isChainedBefore | : Boolean | = false |
| - _previousTaskID | : String | = null |
| - _deadline | : Date | |
| - _priority | : Integer | |

**ITaskWithAttachments**
(traits)

| | |
|---|---|
| + addAttachment (Attachment attachment) | : Attachment |
| + removeAttachment (URL attachmentURL) | : Attachment |
| + getAttachments () | : Collection<Attachment> |

**ITaskWithPriority**
(traits)

| | |
|---|---|
| + getPriority () | : Integer |
| + setPriority (Integer priority) | : void |

**ICompleteReportingTask**
(traits)

| | |
|---|---|
| + getCompleteSOAPAction () | : String |
| + setCompleteSOAPAction (String soapAction) | : void |

**<<Enum>>**
**TaskState**

| | |
|---|---|
| + READY | : EnumConstant |
| + COMPLETED | : EnumConstant |
| + FAILED | : EnumConstant |
| + CLAIMED | : EnumConstant |

**Task**

**ITaskWithInput**
(traits)

| | |
|---|---|
| + getInput () | : Document |
| + setInput (Document input) | : void |
| + setInput (String input) | : void |

**ITaskWithOutput**
(traits)

| | |
|---|---|
| + getOutput () | : Document |
| + setOutput (Document output) | : void |
| + setOutput (String output) | : void |

**ITaskWithState : 1**
(traits)

| | |
|---|---|
| + getState () | : TaskState |
| + setState (TaskState state) | : void |
| + getFailureCode () | : String |
| + setFailureCode (String code) | : void |
| + getFailureReason () | : String |
| + setFailureReason (String reason) | : void |

And below is the HT task instance schema definition:

```xsd
<xsd:complexType name="tTask">
    <xsd:sequence>
        <xsd:element name="id" type="xsd:string" />
        <xsd:element name="taskType" type="xsd:string" />
        <xsd:element name="name" type="xsd:QName" />
        <xsd:element name="status" type="tStatus" />
        <xsd:element name="priority" type="xsd:nonNegativeInteger"
            minOccurs="0" />
        <xsd:element name="taskInitiator" type="htd:tUser"
            minOccurs="0" />
        <xsd:element name="taskStakeholders"
            type="htd:tOrganizationalEntity" minOccurs="0" />
        <xsd:element name="potentialOwners"
            type="htd:tOrganizationalEntity" minOccurs="0" />
        <xsd:element name="businessAdministrators"
            type="htd:tOrganizationalEntity" minOccurs="0" />
        <xsd:element name="actualOwner" type="htd:tUser"
            minOccurs="0" />
        <xsd:element name="notificationRecipients"
            type="htd:tOrganizationalEntity" minOccurs="0" />
        <xsd:element name="createdOn" type="xsd:dateTime" />
        <xsd:element name="createdBy" type="xsd:string"
            minOccurs="0" />
        <xsd:element name="activationTime" type="xsd:dateTime"
            minOccurs="0" />
        <xsd:element name="expirationTime" type="xsd:dateTime"
            minOccurs="0" />
        <xsd:element name="isSkipable" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="hasPotentialOwners" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="startByExists" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="completeByExists" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="presentationName"
            type="tPresentationName" minOccurs="0" />
        <xsd:element name="presentationSubject"
            type="tPresentationSubject" minOccurs="0" />
        <xsd:element name="renderingMethodExists"
            type="xsd:boolean" />
        <xsd:element name="hasOutput" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="hasFault" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="hasAttachments" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="hasComments" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="escalated" type="xsd:boolean"
            minOccurs="0" />
        <xsd:element name="primarySearchBy" type="xsd:string"
            minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax"
            minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
```

*Compare the properties between Tempo PATask with HT task instance scehma definition.

| PATask | Property in Human Task | Comments |
|---|---|---|
| processID | N/A | |
| priority | priority | non negative integer. HT: supports expression evaluation, it is one part of meta data. |
| actionACLs | N/A | HT: According to the role type for the check. Tempo: From the user point to check. |
| output | N/A | |
| date | N/A | Don't know Tempo's data meaning.Need Confirm. |
| description | name? | Need confirm |
| completeSOAPAction | N/A | |
| input | N/A | |
| isChainedBefore | N/A | |
| previousTaskID | N/A | |

| previousTaskID | N/A | |
|---|---|---|
| state | status | "Enumeration. HT: Created, Ready, Reserved, InProgress, Reserved, Completed, Failed, Obsolete, Existed. Tempo: READY, COMPLETED, FAILED, CLAIMED" |
| creationDate | createdOn | |
| attachments | hasAttachments(boolean type) | Tempo's data can be used to indicate this HT's value |
| internalId | N/A | |
| id | id | string type |
| deadline | expirationTime | Are they same? |
| N/A | taskType | HT: Enumeration type, two values are allowed: ""TASK"" and ""NOTIFICATION"" Tempo: Have one Notification class, but it hasn't been mentioned in this document." |
| N/A | taskInitiator | |
| N/A | taskStakeholders | |
| N/A | potentialOwners | |
| N/A | businessAdministrators | |
| N/A | actualOwner | |
| N/A | notificationRecipients | |
| N/A | createdBy | |
| N/A | activationTime | |
| N/A | hasPotentialOwners | |
| N/A | startByExists | |
| N/A | completeByExists | |
| formURL? | presentationName | Need Confirm |
| formURL? | presentationSubject | Need Confirm |
| N/A | renderingMethodExists | |
| output | hasOutput(boolean type) | Tempo's data can be used to indicate this HT's value |
| failureCode, failureReason | hasFault | Tempo's data can be used to indicate this HT's value |
| N/A | hasComments | |
| N/A | escalated | |
| N/A | primarySearchBy | |

**Comments**