Subject: Workflow implementation question
Posted by grim on Mon, 15 Aug 2011 14:15:46 GMT
View Forum Message <> Reply to Message

I have a simple task that adds a new record (add1). On some occasions the data for one particular field will not be known to the user entering the data.

What I would like to do via the workflow is to put an entry on the Workitems list for a user to update the data missing in the recently added record. However, if the particular data had been entered when the record was added, for there to be no record added to the workitems list.

I have tried several ways to achieve this but with no success. It seems that I need to have another transition between my add task (and start place) and the update transition. I would attach my guard statements to this extra transition. One leg would go to the update task and one would go to the end.

Is this a possible way to resolve this issue? What should my extra transition do? Am I going about this in completely the wrong way please can you suggest a better way forward.

With regards

Graham Jones

Subject: Re: Workflow implementation question
Posted by AJM on Tue, 16 Aug 2011 10:43:48 GMT
View Forum Message <> Reply to Message

It sounds like you need to make use of an Explicit OR-split with a guard, as shown in example #4 in Workflow Examples. In your case you need to create a workflow with the following:

1) Your start task is your ADD1 task.
2) There will be one place P1.
3) There will be two transitions:
- T1 (check for missing data), type=automatic
- T2 (add missing data), type=user
4) There will be five arcs:
- Start -> T1, Inward, Sequential
- T1 -> END, Outward, Explicit OR split, Guard= if (strlen($fieldarray['data']) < 1) return TRUE;
- T1 -> P1, Outward, Explicit OR split
- P1 -> T2, Inward, Sequential
- T2 -> END, Outward, Sequential

Transition T1 should be built using pattern UPDATE4. Although it will not actually update the database is has to end with a COMMIT so the the workflow engine knows that the transition has ended. You must ensure that $fieldarray contains the value that will be checked in the guard. For example, if you had multiple fields to check you could create a dummy field, such as

GUARD_CONDITION, and set this to either TRUE of FALSE in your code.

When the workflow is started with the ADD1 task transition T1 will be fired automatically, and the guard will be evaluated. If TRUE the workflow will end, otherwise a token wil be placed on P1 which will wait for transition T2 to be fired. When T2 has been processed the workflow will end.

Subject: Re: Workflow implementation question
Posted by grim on Tue, 16 Aug 2011 13:31:22 GMT
View Forum Message <> Reply to Message

Many thanks for that - I will try it out in a little while.

Cheers

Gj

Subject: Re: Workflow implementation question
Posted by grim on Thu, 25 Aug 2011 17:14:06 GMT
View Forum Message <> Reply to Message

I have been working on my simple workflow and I cannot explain some things. The idea of the workflow is that if one of the data fields is initially left empty, a workflow item is put on the queue to remind the user to complete the data. I hope someone can point me in the correct direction.

Firstly, I set up the workflow as follows:

Insert data task (Add1) -> Place Start

Place Start -> Transition T1 Check for missing data (Upd4)

Transition T1 Check for missing data (Upd4); guard (if field completed) -> Place End
Transition T1 Check for missing data (Upd4); guard (default) -> Place P1

Place P1 -> Transition T2 Add missing data (Upd1); guard(if field completed) -> Place End
Place P1 -> Transition T2 Add missing data (Upd1); guard(default) -> Place P1

This works fine, I run the first task and when it has finished it leaves an item on the workitems queue if the data field is empty. If the field is completed no entries are added to the workitems queue. Clicking on the workitems link launches the update, updating the data ends the workflow. If the user just clicks submit without updating the field, the item remains on the queue.

I have put some debugging statements into _cm_pre_updateRecord to see what is in fieldarray. I have a two dimensional array, the first being the whole record of which there is only one. The second array is associative and contains each of my data fields. As I expected, it looks exactly the

same whether it comes from the Upd4 process or the Upd1 process.

In my guard statement for the Upd4 process I have to refer to my data field as:

$fieldarray[0]['cref']

I expected this based on the above debug info. However, when I use the same statement in the Upd1 guard statement, it doesn't find my data. To make it work, I have to refer to my data field as:

$fieldarray['cref']

I cannot get even close to working out why the two statements are different especially when the format of the fieldarray is the same when in the _cm_pre_updateRecord function.

I hope that someone can answer this question for me.

As always, thanks in advance.

Regards

Graham Jones

Subject: Re: Workflow implementation question
Posted by AJM on Thu, 25 Aug 2011 17:44:27 GMT
View Forum Message <> Reply to Message

This is because the UPDATE4 pattern can deal with multiple records, so $fieldarray is indexed by record number. The UPDATE1 pattern can only deal with a single record, therefore $fieldarray is not indexed by the record number.

The _cm_pre_updateRecord method works on a single record record at a time. When the pattern deals with multiple records it is called separately for each record, not once for all records.
