

# Workflow Scenario: Trouble Ticket (Version 2)

---

---

Copyright 1998 Netscape. This document is provided by Netscape Communications Corporation to the members of the OMG for the purpose of evaluating workflow. Permission is hereby granted for copying and redistribution of this document as long as this copyright notice remains in place.

---

---

## Setting

The trouble ticket scenario covers quality assurance teams or customer support teams. A "bug" or "problem" is identified; it must be recorded; the record must be checked for accuracy; from a single instance of a problem, the underlying cause is identified; a resolution is identified, which must be communicated back to the original party with the problem.

The scenario has been modified in version 2 to provide for a subprocess invocation in order to be able to discuss the interoperability of two workflow systems. Also an appendix has been added to explain how the JFlow interfaces would be used to implement this.

## The Process

First we will present a process centric view of what happens to a given trouble ticket

### Step 1: Recording the Problem -

The problem may be found by an internal or external person. There are two ways that a problem can come from the external person: by phone or by email.

For the internal person, there must be a screen that can be called up without delay that presents a form for entering the details of the problem. Submitting this form will cause the creation of the workflow process, and at the same time generate a unique ID for the trouble ticket.

A phone call from an external person will be handled by an internal person, who uses essentially the same form mentioned above, but needs to be able to search for registered customers a couple of different ways. Again, submission of the form will create the process and assign an id. The ID of the trouble ticket must be produced by the system and be immediately available (within 10 seconds) in order to let the external person know the trouble ticket ID. The ID is used as a way to call up the trouble ticket when that person calls in again to check on progress.

Finally, email may be sent to a particular address. This email is automatically picked up, and a trouble ticket started, which includes the body of the email as part of the data. No attempt is made to automatically analyze the body of the message, but the sender's email address is retrieved from the header. The first step of the process is for some internal person to read the message, and to fill in some of the other fields on the form with real values, and then submit it. The system should be able to look up the registered user from the email address. When the trouble ticket is submitted, a email confirmation is sent to the external person, and the process goes to step 2.

### Step 2: Reproduce the Problem

This step is designed to check the trouble ticket report, and to see if it describes accurately a

reproducible problem. This activity is simply to follow the instruction on the report, and to see if the described behavior occurs. If the trouble ticket comes from an internal person, then this activity must be assigned to someone else so that the recording and reproducing of the problem are not done by the same person. If it comes from an external person, this activity may be done by the same person who enters the report. If the behavior can not be reproduced, then this process goes to step 3, otherwise it goes to step 4.

The problem may be identified at this stage. If there is a known solution to the problem it should be entered or referred to at this stage, and then communicated back to the originator by going to Step 6. If the problem is recognized as a duplicate of another problem, it should be able to be recorded as such, and go to step 5.

### **Step 3: Correcting the Report**

This step is reached only if the problem can not be reproduced. This step is assigned to the originator if internal. If external, this must be assigned to a person who can contact the originator and get more clarification on the problem. There are two results of this step, either back to Step 2, or to give up on the process and go to step 6.

### **Step 4: Identifying the Problem and Resolution**

This is where the specialist is called in. The problem details should narrow down the area of the problem. If the expert determines that the area is wrong, it should be able to be reassigned, and the person assigned to the activity should change. The problem stays in this state until a resolution is determined. Either the problem is identified and it will be fixed, or it be fixed later due to schedule constraints, or it is determined to be a misunderstanding and is actually the correct behavior. In all cases the resolution must be communicated to the originator, either via email, or else through a phone call. It goes to step 5.

For this organization, accomplishing this activity require invoking a subprocess. The development team has its own workflow process that handles this in a manner that fits the way they work. The exact route of this sub process is not the subject of this scenario, only that it is started, it is given data, and at some time later it reports that it is complete and returns a set of data.

The subprocess for the development team was implemented before the trouble ticket scenario, so it already has a set of field with meaning appropriate to that task. This means that the trouble ticket process must translate the fields into the field used by the sub process. The details of this is defined below.

### **Step 5: Verifying the Resolution**

When the problem is resolved, then it waits for the resolution to become available. When it is available, the resolution is verified. If the resolution was "fixed," and the problem is not actually solved, then the process can be sent back to step 4. Otherwise the process goes to step 6.

### **Step 6: Communicate Results**

The results of the process are communicated back to the originator here. This step contains a rule that the result must be communicated within 3 days of being known. If not, an email message is

sent to the support manager.

## Step 7: Audit and Record

This step may happen before or after Step 6, but must happen before the end of the process. It involves someone looking at the process and determining whether the question/answer should be included in a monthly newsletter to the user community. This step is started in parallel with Step 6 since it does not depend upon it, and might happen before it.

## The Data

This first set of data is the set of fields used within the Trouble Ticket process itself.

- Originator UID - a unique id if one exists
- Originator Name
- Originator Address
- Originator phone
- Originator email address
- Submitter - the person who took the call. If internal, same as originator.
- Synopsis - a one line description
- Description - a full indepth description
- Source Email - holds the email that started the process, if there is one.
- Severity
- Priority
- Product
- Area
- Date Received (Submitted)
- Date Resolved
- Date Verified
- Date Closed
- Attached data files (URLs to files checked into a server)
- Expert - the person who is the expert for the current product area.
- Resolution
- Resolution Description
- Status (presumably part of the workflow...)
- Date of last originator contact
- Duplicate Ticket number.

This second set is the data needed by the subprocess invoked under step number four.

- ShortDescription - a one line description of the problem (like Synopsis above)
- QAPriority - the priority assigned by QA (like Priority above)
- Priority - the priority within the development team. (not present in the parent process)
- Description - the full description of the problem (like Description above)
- Attached files (like above)
- Submitter - (like above)
- SubmittedDate - the date submitted to the development team.
- A number more data fields internal to this process.
- Resolution - this is a result that is passed back to the parent process.

## Interaction Scenario

### Day 1

(a) An important customer calls up with a problem. Person A takes the call, and brings up the form, enters the information, submits it, and sees that it is currently assigned to Person B. Later person B sees it on the worklist, but does not call it up.

### Day 2

(b) Person A checks on the status of the trouble ticket (where was this listed, since it is not on his worklist?), and sees that Person B has not taken any action. Person A composes an email message to Person B, with the process status page attached.

(c) Person B receives the email, clicks on the link to call up the process in the browser. Person B reproduces the problem, makes sure it is correctly assigned to Product X, Area 1, and then completes the activity. It disappears from B's worklist.

(d) Step 4 becomes active, and it looks up in the directory to find the workflow process definition that is responsible for Product X, Area 1. It does not matter that the definition is in a different workflow engine, an instance of it is created. The data is passed to it. The subprocess determines that for Product X Area 1 the first activity on the subprocess should be assigned to user C. It appears on C's worklist, and C has specified to receive notification by email, so it sends him an email message.

### Day 3

(e) Person C gets the email, and clicks on the link to bring up the sub process instance, looks at it for 5 minutes, and determines that it should be in Area 2. He changes the Area field to 2. The system finds out that person D is assigned to Area 2, so it disappears from C's worklist, appears on D's worklist, and D also receives an email notification.

(f) Person A checks on the progress of the case and sees that step 5 is currently active. He sees that it has a subprocess. He can navigate to the subprocess and see what step it is in there and that it is currently assigned to person D. He can read the output data from the subprocess which includes the resolution of the problem, which at this time is still empty.

### Day 4

(g) Person D gets the message, retrieves the process instance, but does not have the time to get to it. Instead he determines that Person E should be able to handle the task, so he delegates the activity to Person E.

(h) The important customer calls up again but this time to the Vice President of the division, who talks to Person A, who checks on the status. Person A raises the priority data field in the process in response.

(i) Person E sees it immediately, and gets to work. A short time later a solution is found; he enters the resolution, and marks it fixed, completing the subprocess, passing the data back to the main

process, and finally ending up at step 5. This is assigned to Person B since B was the one who reproduced the problem. B verifies the fix, and the process goes to step 6. This is assigned to Person F.

#### Day 5 & 6

(j) Nothing happens to Step 6

(k) The newsletter editor picks up and completed Step 7, removing it from his worklist.

#### Day 8

(l) Still nothing has happened. There is a rule that the result must be communicated back within 3 days of the fix, so a trigger goes off, causing an email message to be sent to Person A, and to Person G who is the manager of Person F.

(m) Person G knows that Person F is on vacation, and forgot to set up automatic forwarding of responsibility. Since this activity needs to be attended to, reassigns the task to person H (how did he claim authority to do so?) who gets an email notification, and takes case of the activity completing the process.

#### Day 9

(n) Person A checks back in on the process. The process is finished, but A is still able to access the history and final state of the process.

## jFlow Interface use in Scenario

This section discusses how the jFlow interfaces are used to implement this scenario. First thing that should be mentioned is that jFlow is supporting the interoperability of workflow, so it is only used when communicating between workflow processes. A lot of things are internal to the process system, such as determining who should do a task and causing the work item to be created.

a: Person A uses a tool that knows about the process directory object. From this, it selects the process definition needed. Then an instance of the process is created. This process definition could be consulted to get the list of data items that the process definition is expecting to receive at startup time. The process is started through the WorkProcess interface. Person B checks his worklist by searching for all WorkItem objects which as assigned to him. The system has created the workitem through the use of internal functionality. The search is a function of the Framework.

b: Person A brings up a tool that accesses ProcessDirectory, then the ProcessDefinition where he can search for WorkProcesses that he started or could search using any number of other criteria to find the WorkProcess object that represents this process instance. From the process instance he can ask for the current activities and see that it is still in step number 2. By retrieving the activity object, he can request who is assigned to it, and see that it is person B. Using an external phonebook he looks up that person's email address for the email message.

c: Person B has been given a link to the process instance, presumably using a Corba name through which he retrieves the WorkProcess object, from that the activity object, and using getContext all

the data from the process instance which includes all the details of the customer call. (Strictly speaking he retrieve only those fields which the designer of the process determined was needed to do this activity, which might, in this case, be all the data.) This includes empty fields for the product and area which is filled in. setResult is used to put the values in. The complete method on the activity is used to indicate that the activity is done. The subsequent logic that removes this item from the worklist (deleting the workitem object) is internal functionality not exposed by jFlow.

d: A custom functionality is used to associate the product and area with a process definition object. The subprocess has a different set of fields, some of which are the same meaning with different names. Th designer of the process must have defined the following mapping:

- Synopsis <--> ShortDescription
- Product <--> Product
- Area <--> Area
- Priority <--> QAPriority
- (no field) <--> Priority
- Description <--> Description
- Attached Files <--> Attached files
- Submitter <--> Submitter
- (no field) <--> SubmittedDate
- Resolution <--> Resolution

This mapping and the way that the mapping is specified is accomplished with internal proprietary means. The only externally observable indication that this mapping has taken place is that a data object is available with the right field names and values. The ProcessDefinition.getContextInfo method might be used to verify that all the fields in the generated data object as suitable for the subprocess, but the mapping can not be generated from this info. Ultimately, the createWorkProcess method is called, passing the activity as the requester, and the newly constructed data object as the context of the new process. A reference to the new sub process instance is saved in the WorkActivity object for future reference. This, the creation of the WorkItem, and the sending of the email is internal functionality that is not accomplished through jFlow interfaces.

e: Person C has been sent a name of the process instance object which is retrieved through a name service. From the process instace (WorkProcess) he finds the WorkActivity object, from which he can retrieve the current context data for the process. After reading it, he sets the Product and Area fields using the setResult method. This assumes that these fields are in both the input and the output sets. The process has some internal logic to determine the person from the product area, and it must have some trigger that tells it to reevaluate when these field values change. This functionality is not accomplished with jFlow interfaces. The result is that the activity is assigned to Person D. Internal functionality deletes the WorkItem for person C, and creates one for Person D. Also, the subprocess sends a dataChangedEvent to the requesting process updating it about the Product and Area fields.

e\*\*: There is a more complex possibility. If area 2 is suported by a different process definition, then the requesting process presumably responds to the receipt of the dataChangedEvent by determining what new process definition should be used, terminating the current sub process by calling the terminate command on the WorkProcess, and then creating a new process from the new ProcessDefinition and a newly formed context data object. Determining that this is the right thing to do requires fairly sophisticated capability from the process model, but the point of mentioning it

here is simply to show that such operations can be accomplished through the jFlow interfaces.

f: Person A finds the original WorkProcess object, from this finds the current activity, from this uses getPerformer to get to the sub process, then the activ activity is found again, and the person assigned to it, as well as using getResult to check the current data values.

g: Person D starts with the WorkProcess Object, gets the current activity, reviews the context data, and uses the reassign command on the activity to give it to Person E.

h: Person A retrieves the main process instance in the same manner as before, reviews the data using getResult, and updated the priority using setContext. Because the priority is mapped to the sub processes QAPriority, the main process calls the subprocesses setContext method with the new value for QAPriority.

i: Person E retrieve his worklist (a search for WorkItems assigned to him) from this retrieves the WorkActivity, and from this the WorkProcess. From the activity he uses getContext to view the data values. He updates the Resolution field using the setResult method on the activity. This causes the data to be propogated to the parent process via a dataChangedEvent. He also calls completed on the activity. The scenario allow for any number of sub steps. Eventually one will cause the sub process to be completed. This causes the sub process to call the completed function on the WorkActivity representing Step 4, and passing the final result values. These values are mapped back into the values for the main process. Since step 4 is completed, step 5 is activated and appropriate things are done to worklists.

j: nothing happens

k: no different than previous such accesses except that when retrieving the current activities (getContainees) returns two activity object (Step 6 and Step 7) and the user must choose the one that is assigned to him.

l: the sending of the email is an internal functionality that does not use jFlow interfaces

m: Person G accesses the WorkProcess and from that the WorkActivity. Uses the reassign function to assigne Person H tothe activity. The system updates the worklists. Person H assesses the WorkProcess, then the Activity, then calls complete on the activity. This causes the process to end.

n: Person A searches history using the getHistory function on the ProcessDirectory. The process instance object no longer exists, but history is still accessible.

---

Feb 9, 1997, Keith D Swenson, Netscape Communications Corporation