

Supporting Document

Workflow Scenario: Trouble Ticket

Hitachi, Ltd.

Supporting Document Contact Points

Toshiaki Sakaguchi, t-sakagu@sdl.hitachi.co.jp

Shunsuke Akifuji, akifuji@sdl.hitachi.co.jp

Table of Contents

1.	<i>Introduction</i>	2
2.	<i>Scenario Realization</i>	3
2.1	Evaluation Criteria	3
2.2	Use Cases Suggested from the Scenario	3
2.3	Use Cases Realization by joint Workflow Spec	5
2.4	Other Requirements Suggested from the Scenario	9
3.	<i>Conclusion</i>	10
4.	<i>References</i>	10

1. Introduction

The purpose of this document is to illustrate how the Workflow Management Facility joint submission [1] can be used to realize the workflow scienario(version 2) provided by Keith D. Swenson, Netscape Communication Corporation[2].

2. Scenario Realization

2.1 Evaluation Criteria

We analyze the scenario using use case for extracting system requirements. Then we evaluate the realization for each use case.

There are seven activities in the process definition. Each activity corresponds to the step of the scenario. When each step completes, the next step will be decided based on the attributes of the step.

2.2 Use Cases Suggested from the Scenario

For evaluating joint Workflow submission, We should identify the scenario. That is to say, we should describe the scenario by UML use cases. Then we extract actors and use cases from the scenario.

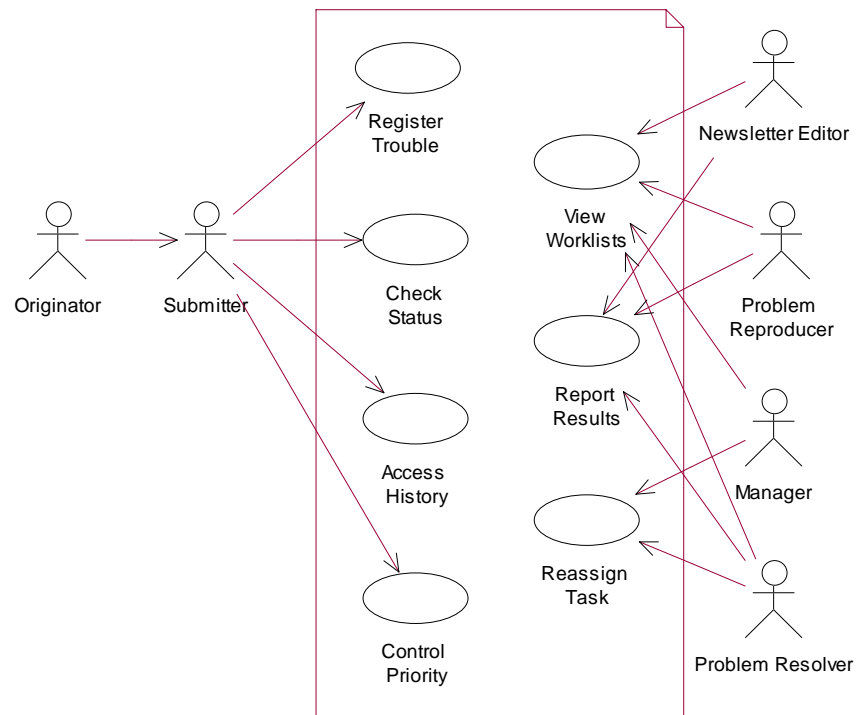


Figure 1 Trouble Ticket Use Case Diagram

2.2.1 Actors

2.2.1.1 Originator

Originator is a person who reports problems from external. He/she reports problems to *Submitter*, who is a internal person.

2.2.1.2 Submitter

Submitter is a person who reports problem from internal. He/she delegates *Originator* for registering problems to the system.

2.2.1.3 Problem Reproducer

Problem Reproducer is a person who reproduces the problem *Submitter* reports. If the problem is from *Originator*, *Submitter* can be *Problem Reproducer*.

2.2.1.4 Problem Resolver

Problem Resolver is a person who analyzes the problem and finds a resolution.

2.2.1.5 Manager

Manager is a person who manages work status of persons. He/she reassigns tasks if needed.

2.2.1.6 Newsletter Editor

Newsletter Editor is a person who collects problems and resolutions arised in one month.

2.2.2 Use Cases**2.2.2.1 Register Trouble Ticket**

This use case describes registering a new trouble ticket. It appears at Step 1 in the process and Day 1 (a) in the interaction scenario. This case has following subflows:

- bring up the form
- enter the information
- submit the form

2.2.2.2 View Worklist

This use case describes checking work assigned to the users. It is used at Day 1 (a), Day 2 (c)(d), and Day 3 (e). All Actors who use this system use this use case.

2.2.2.3 Check Status

This use case describes checking the status of registered trouble tickets. Judging from the scenario, the originator of the ticket has a duty of checking if the process is working. This is used at Day 1 (b), Day 3 (f), and Day 4 (h). *Originator* usually uses this use case in the scenario.

2.2.2.4 Access History

This use case describes surveying the trouble ticket. It is used at Day 9 (n). *Originator* usually uses this use case in the scenario.

2.2.2.5 Reassign Task

This use case describes reassigning a task to other user. If the user, who is assigned to the task, is in difficult situation to do it, he or his manager will have a duty of reassign the task to the proper person. It appears at Day 3 (e) and Day 4 (g).

This use case is used by a person who task is assigned to , and a manager of the person.

2.2.2.6 Control Priority

This use case describes changing the priority of the trouble ticket. It is used at Day 4 (h). *Originator* uses this use case.

2.2.2.7 Report Results

This use case describes setting results to a trouble ticket case. The contents of the result depend on the process. For example,

- result if the problem is reproduced or not(Step 2)
- result of the problem categorization(Step 2)
- result of the problem resolution(Step 4)
- result of the resolution verification(Step 5)
- result of the newsletter adoption(Step 7)

2.3 Use Cases Realization by joint Workflow Spec**2.3.1 Register Trouble Ticket**

Bringing up form and entering information are GUI specific procedure. The metadata for making forms are got from WfProcessMgr operation context_signature() and result_signature(). For submitting the form, operation set_process_context() and start() are used.

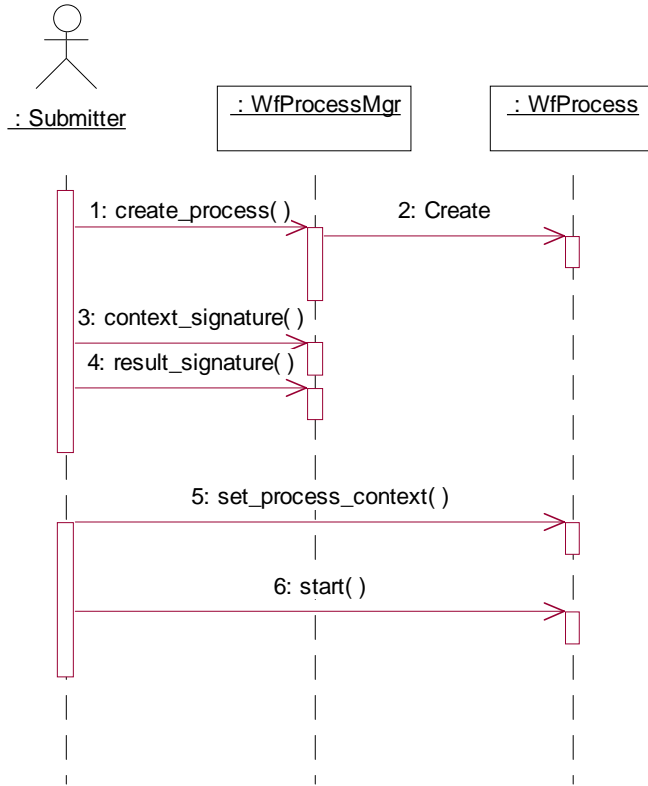


Figure 2 Register Trouble Ticket Sequence Diagram

2.3.2 View Worklists

For viewing worklists, user gets reference of WfResource using Naming Service or something, and gets his worklist using get_iterator_work_item or get_sequence_work_item operation. Worklist is expressed as WfAssignment.

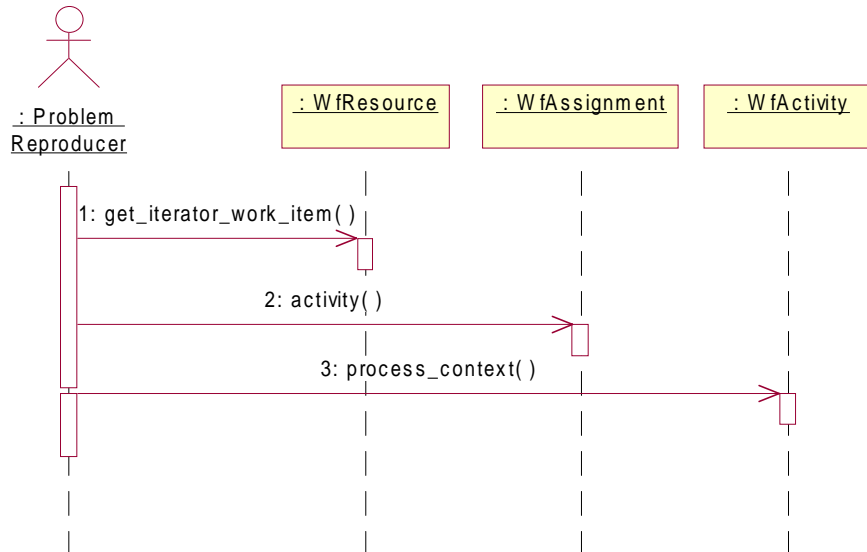


Figure 3 View Worklists Sequence Diagram

2.3.3 Check Status

For checking the status of the submitted case, user invokes `get_iterator_step()` operation and gets a series of `WfActivity` object contained in the process. Then user can check status by `state()` operation, or check assignment status by `get_iterator_assignment()` operation.

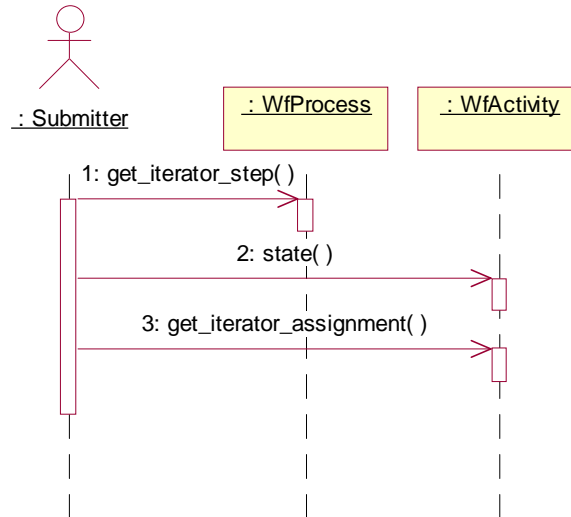


Figure 4 Check Status Sequence Diagram

2.3.4 Access History

For accessing the history of the submitted case, user gets reference of `WfProcess` which inherits `WfExecutionObject`. And user invokes an `WfExecutionObject` operation `get_history_iterator()`, which gets the reference of `WfEventAuditIterator`. User can browse the history of the case by operation `get_next_object()`, `get_previous_object()` and so forth.

User can access history within the `WfProcess` object lifecycle. So, If the object is destroyed after the completion, user requires the reference of `WfEventAudit` to know the history.

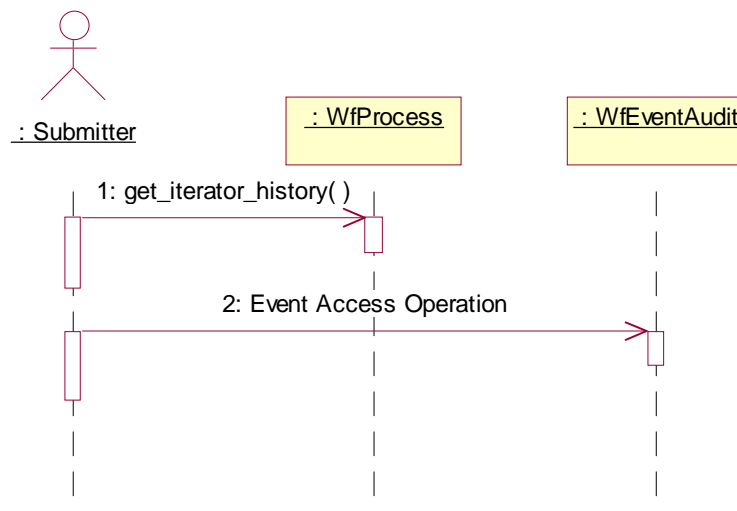


Figure 5 Access History Sequence Diagram

2.3.5 Reassign Task

For reassigning the task to other users, user releases the task assignment using `release()` operation, and gets the reference of `WfResource` object which correspond to the target user, and invokes a `WfAssignment` operation `set_assignee()`.

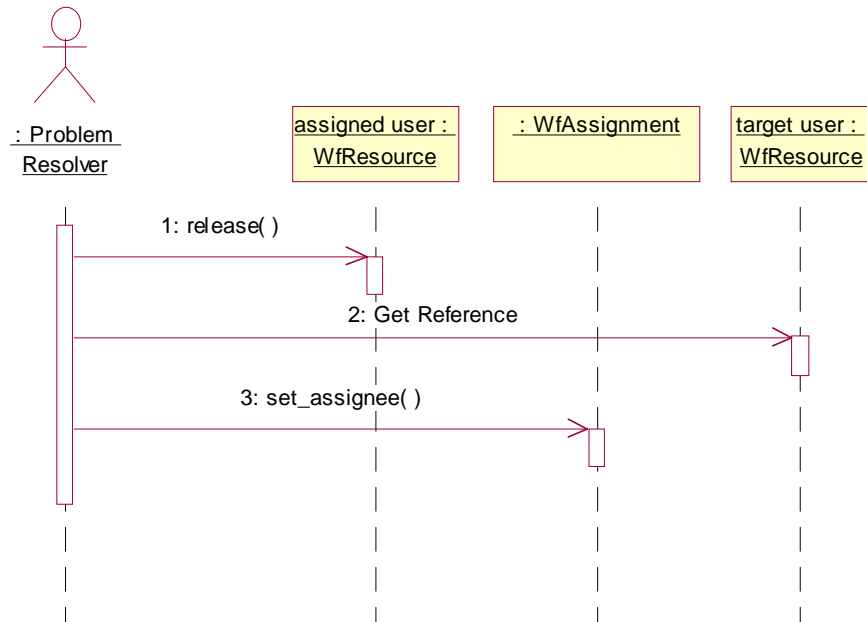


Figure 6 Reassign Task Sequence Diagram

2.3.6 Control Priority

For controlling the priority of case, user gets a reference to `WfProcess` Object corresponds to the case, and invokes `set_priority()` operation. Priority change affects the priority of `WfActivity` Objects contained in the process, and the person assigned to the activity can see it.

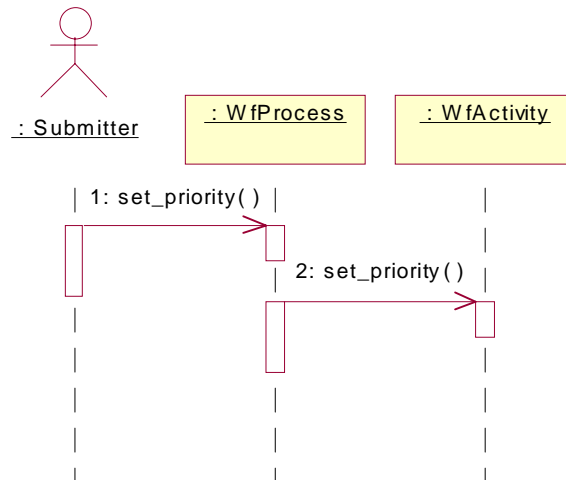


Figure 7 Control Priority Sequence Diagram

2.3.7 Report Results

For reporting results, user gets a reference to `WfActivity` object and set results by `set_result` operation. If the task is finished, user can send `complete` to the activity.

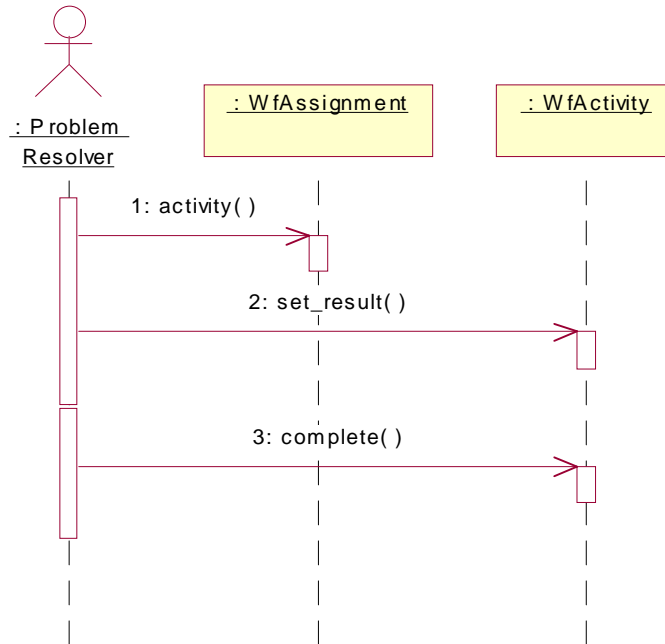


Figure 8 Report Results Sequence Diagram

2.4 Other Requirements Suggested from the Scenario

2.4.1 Subprocess Coordination

In the scenario, Step 4 task is delegated to other section (the development team). And the team has its own workflow process. Subprocess is invoked by WfActivity object.

WfActivity object, which inherits WfRequester, searching an appropriate WfProcessMgr object, submits a process, and gets results by set_results operation.

2.4.2 Notification to Users

In the scenario, if a task is assigned to someone, the system automatically sends e-mail for notification to him. This process can be realized by watching a status change of WfResource or WfAssignment. WfResource and WfAssignment inherit BaseBusinessObject, which extends CosNotifyComm::StructuredPushSupplier. So the application consuming the status change event and invoking e-mail is feasible.

2.4.3 Data Description

All data are stored into ProcessData struct. ProcessData struct is a series of NamedValue struct, which consists of the name of value and value itself.

3. Conclusion

Joint Workflow spec can implement the Trouble Ticket scenario from the point of execution environment. But in case of user accessing the history after the process terminates, there would be better if there exist events retrieving function from the key of the process, because user has no way of getting the reference of WfEventAudit object corresponds to the process.

4. References

1. Joint Workflow RFP Revised Submission, OMG Document bom/98-06-07
2. Keith D. Swenson, Netscape Communications Corporation, Trouble Ticket Workflow Scenario (Version 2), OMG Document bom/98-02-09.