Automated Reduction

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Brief Description 5

2. Preconditions 5

2.1 < Precondition One > 5

3. Basic Flow of Events 5

3.1 Incoming Message – Request to Reduce Data Set 6

3.2 Retrieve Data Set Meta Data 6

3.3 Determine Appropriate Workflow(s) 6

3.4 Execute Workflow(s) 6

3.5 Send Message to Reducer 7

3.6 Process/Handle Response Message From Reducer 7

3.7 Next Run Set Workflow Description Task - Stich Runs Together 7

4. Alternative Flows 8

4.1 <Area of Functionality> 8

4.1.1 < A1 First Alternative Flow > 8

4.1.2 < A2 Second Alternative Flow > 8

4.2 <Another Area of Functionality> 8

4.2.1 < AN Another Alternative Flow > 8

5. Subflows 8

5.1 <S1 First Subflow > 8

5.2 < S2 Second Subflow > 8

6. Key Scenarios 9

7. Key Assumptions 9

8. Postconditions 9

8.1 < Postcondition One > 9

9. Extension Points 9

9.1 <Name of Extension Point> 9

10. Special Requirements 10

10.1 < First Special Requirement > 10

11. Additional Information 10

11.1 Current Unknown Information 10

11.1.1 Message from Translation service 10

11.1.2 Logging and Persisting of Workflow State 10

12. Information Gathering: 11

12.1 Messages 11

12.1.1 Transition Service 🡪 Workflow System: POSTPROCESS.DATA\_READY 11

12.1.2 CATALOG.DATA\_READY 11

12.1.3 REDUCTION.DATA\_READY 11

12.1.4 REDUCTION\_CATALOG.DATA\_READY 11

12.2 Meta Data 11

12.3 WorkflowDescription(s) 11

[The following template is provided for a Use-Case Specification, which contains the textual properties of the use case. This document is used with a requirements management tool, such as Rational RequisitePro, for specifying and marking the requirements within the use-case properties.

The use-case diagrams can be developed in a visual modeling tool, such as Rational Rose. A use-case report, with all properties, may be generated with Rational SoDA. For more information, see the tool mentors in the Rational Unified Process.]

# Brief Description

[The description briefly conveys the role and purpose of the use case. A single paragraph will suffice for this description.]

This use case begins when the Translation Service sends a message to the Workflow System that a new data set is ready to be reduced. On reception of the message the Workflow system must determine the context of the data set from the meta data, determine which workflow is to be used to reduce the data set. The Workflow system does not actually do any of the processing of the data set. Instead the Workflow System sends message to reducer systems (post processing systems) to do the actual reduction

# Preconditions

[A precondition of a use case is the state of the system that must be present prior to a use case being performed.]

## < Precondition One >

# Basic Flow of Events

[This use case starts when the actor does something. An actor always initiates use cases. The use case describes what the actor does and what the system does in response. It is phrased in the form of a dialog between the actor and the system.

The use case describes what happens inside the system, but not how or why. If information is exchanged, be specific about what is passed back and forth. For example, it is not very illuminating to say that the actor enters customer information if it is not defined. It is better to say the actor enters the customer’s name and address. A Glossary of Terms (or a more formal Domain Model) is essential to keep the complexity of the use case manageable⎯you may want to define things like customer information there to keep the use case from drowning in details.

Simple alternatives may be presented within the text of the flow of events. If it only takes a few sentences to describe what happens when there is an alternative, do it directly within the flow. If the alternative flow is more complex, use a separate section to describe it. For example, an **Alternative Flow** subsection explains how to describe more complex alternatives.

Complex flow of events should be further structured into sub-flows. In doing this, the main goal should be improving the readability of the text. Subflows can be invoked many times from many places. Remember that the use case can perform subflows in optional sequences or in loops or even several at the same time..

A picture is sometimes worth a thousand words, though there is no substitute for clean, clear prose. If it improves clarity, feel free to paste flow charts, activity diagrams or other figures into the use case. If a flow chart is useful to present a complex decision process, by all means use it! Similarly for state-dependent behavior, a state-transition diagram often clarifies the behavior of a system better than pages upon pages of text. Use the right presentation medium for your problem, but be wary of using terminology, notations or figures that your audience may not understand. Remember that your purpose is to clarify, not obscure.]

From a black box perspective the interactions are rather limited. However, it is the internal algorithm for establishing and processing the workflow that is critical. The following represents the high level steps that are expected of the Workflow System to process the reduction of a data set.

This happy day scenario is for the Run Set to process a Run and stitch it together with any other existing runs.

Use Case Steps:

## Incoming Message – Request to Reduce Data Set

The Translation Service sends a message to Workflow System

The Workflow System identifies the message as a request (command) to reduce (process) a raw data set.

## Retrieve Data Set Meta Data

The Workflow System retrieves the meta data about the raw data set.

Note: It is not clear where the meta data is or where it is retrieved from. Simplistically it would be included as part of the received message. However, it could be that the meta data must be retrieved from some other place.

## Determine Appropriate Workflow(s)

Using the Meta Data the Workflow System determines the Context of the data set.

The contents of the context are TBD, but are expected to include: the ID of the instrument that produced the data set, the experiment the data set belongs to, the run set the data set belongs to

Using the Meta Data the Workflow System determines which Experiment the Data Set is part of. If the Experiment does not yet exist, the Workflow System creates one.

Using the Meta Data the Workflow System determines which Run Set the Data Set is part of. If a Run Set does not yet exist, the Workflow System creates one and associates it with the Experiment

If the Run Set needs to be created, the Workflow System determines which Procedure (recipe) to associate with the Run Set. Then the Workflow System creates a Workflow that binds (associates) the Procedure with the Run Set.

Using the Meta Data the Workflow System determines which Procedure to associate with the Run (Data Set). Then the Workflow System creates a Workflow to bind the Run (Data Set) and the Procedure.

The Workflow System associates the Run Workflow with the Run Set Workflow. (The Run and the Run Set should also be associated together.)

[The following UC steps are notional at this point. Specific workflows and actions need to be established.]

## Execute Workflow(s)

[Loop through the following steps until the Run Workflow and the RunSet workflow have been completed]

The Workflow System looks at the Run Set Workflow to determine what action to take. In this scenario the action is to reduce the Run

The Workflow System looks at the Run Workflow to determine what action to take. In this scenario the action is to send a specific message to a Reducer. (Note the specific message is TBD.)

The Workflow System creates a message to send to the Reducer requesting a specific reduction action and all the information it will need to perform the requested action. (Note this is extremely vague and needs to be made precise.)

## Send Message to Reducer

The Workflow System sends the message to the Reducer.

The Reducer receives the message and processes the action. Note that since the Reducer is an actor and is external to the Workflow System, what processing is done and how the reduction is done by the reducer out outside the scope of the Workflow System and this document.

The Reducer sends a message to the Workflow Systems indicating the requested action is complete. (Note: currently this message and its contents are undefined. It must at least have enough information to allow the Wrokflow System to determine what Run (Data Set) the message is related to.)

## Process/Handle Response Message From Reducer

The Workflow System receives the incoming message from the reducer.

From the incoming message the Workflow System determines which Run (Data Set) the incoming message is related to.

The Workflow System determines if the incoming message satisfies the current step of the Run workflow. If it does not, do nothing. If it does get the next step in the Run Workflow. Again the next step, as will all Run Workflow Steps is to send a message to a reducer.

[Loop above steps until all steps in the Run Workflow have been completed]

When the Run Workflow is complete, the Workflow system checks if the current step in the Run Set Workflow has been successfully completed. For this scenario we will say yes because the Run Workflow is complete and there are no other run workflows to process or that are in processing.

## Next Run Set Workflow Description Task - Stich Runs Together

The Workflow system then gets the next step in the Run Set Workflow. For this scenario it is to stitch the current run together with any other runs.

The Workflow Systems creates a message to a Reducer to stitch together the current Run (reduced Data Set). (TBD – what does this message need to contain? Where is the reduced run?)

The Workflow System sends the message to a Reducer.

The Reducer processes the message and stitches together the current run with other runs. Since this is external to the Workflow System it is outside the scope of the system and this document.

The Reducer sends a message to the Workflow System indicating the stitching is complete. (Note: this message is TBD.)

On receiving the message from the reducer the Workflow System identifies the Experiment and Run Set the message is related to.

The Workflow System determines if the Run Set Workflow step is complete or not.

If it is complete the Workflow System determines if the Run Set Workflow Procedure is complete.

If the Run Set Workflow Procedure is complete do nothing and the use case ends.

# Alternative Flows

[More complex alternatives are described in a separate section, referred to in the **Basic Flow** subsection of **Flow of Events** section. Think of the **Alternative Flow** subsections like alternative behavior⎯ each alternative flow represents alternative behavior usually due to exceptions that occur in the main flow. They may be as long as necessary to describe the events associated with the alternative behavior.

Start each alternative flow with an initial line clearly stating where the alternative flow can occur and the conditions under which it is performed.

End each alternative flow with a line that clearly states where the events of the main flow of events are resumed. This must be explicitly stated.

Using alternative flows improves the readability of the use case. Keep in mind that use cases are just textual descriptions, and their main purpose is to document the behavior of a system in a clear, concise, and understandable way.]

## <Area of Functionality>

[Often there are multiple alternative flows related to a single area of functionality (for example specialist withdrawal facilities, card handling or receipt handling for the Withdraw Cash use case of an Automated Teller Machine). It improves readability if these conceptually related sets of flows are grouped into their own clearly named sub-section. ]

### < A1 First Alternative Flow >

[Describe the alternative flow, just like any other flow of events.]

#### < An Alternative Subflow >

[Alternative flows may, in turn, be divided into subsections if it improves clarity. Only place subflows here is they are only applicable to a single alternative flow.]

### < A2 Second Alternative Flow >

[There may be, and most likely will be, a number of alternative flows in each area of functionality. Keep each alternative flow separate to improve clarity.]

## <Another Area of Functionality>

[There may be, and most likely will be, a number of areas of functionality giving rise to sets of alternative flows. Keep each set of alternative flow separate to improve clarity.]

### < AN Another Alternative Flow >

# Subflows

## <S1 First Subflow >

A subflow should be a segment of behavior within the use case that has a clear purpose, and is "atomic" in the sense that you do either all or none of the actions described. You may need to have several levels of sub-flows, but if you can you should avoid this as it makes the text more complex and harder to understand.

## < S2 Second Subflow >

[There may be, and most likely will be, a number of subflows in a use case. Keep each sub flow separate to improve clarity. Using sub flows improves the readability of the use case, as well as preventing use cases from being decomposed into hierarchies of use cases. Keep in mind that use cases are just textual descriptions, and their main purpose is to document the behavior of a system in a clear, concise, and understandable way.]

# Key Scenarios

[List the most important scenarios of the use case. Simply provide a short name and accompanying description to uniquely identify each key scenario. There will potentially be many scenarios possible with this use-case specification: it is important to focus on the most important or frequently discussed scenario’s that are either exemplars of this use case or are of concern or specific importance to the actor stakeholders.]

# Key Assumptions

[A precondition of a use case is the state of the system that must be present prior to a use case being performed.]

There are some key assumptions being made that about…

A Data Set represents a collection of information that has been generated by an instrument. For the purposes of this system the data set represents a bounded run of an experiment. Each run is a new instance of a data set.

A Run belongs to a Run Set. A Run set of a set of runs that are grouped based on some purpose. For instance, there could be a Run Set for that is the result of calibrating the instrument. There is going to be a Run Set that represents the raw data produced for the experiment. Each Run (Data Set) belongs to only one Run Set.

A Run Set, and hence a Run is part of an Experiment.

A Data Set has a set of Meta Data that provides information about the Data Set.

The Meta Data is expected to identify:

* The Experiment the Data Set belongs to
* The Run Set the Data Set belongs to
* Some information that can be used to determine which Procedure (recipe) to use to process (reduce) the Data Set. THIS IS TBD AS IT IS CURRENTLY UNKNOWN.

A Reducer is an Actor to the Workflow System that is responsible for processing the messages that direct the reduction of a data set.

# Postconditions

[A postcondition of a use case is a list of possible states the system can be in immediately after a use case has finished.]

## < Postcondition One >

# Extension Points

[Extension points of the use case.]

## <Name of Extension Point>

[Definition of the location of the extension point in the flow of events.]

# Special Requirements

[A special requirement is typically a nonfunctional requirement that is specific to a use case, but is not easily or naturally specified in the text of the use case’s event flow. Examples of special requirements include legal and regulatory requirements, application standards, and quality attributes of the system to be built including usability, reliability, performance or supportability requirements. Additionally, other requirements⎯such as operating systems and environments, compatibility requirements, and design constraints⎯should be captured in this section.]

## < First Special Requirement >

# Additional Information

[Include, or provide references to, any additional information required to clarify the use case. This could include overview diagrams, examples or any thing else you fancy.]

## Current Unknown Information

### Message from Translation service

Need to figure out what this is, what it looks like, what its format is, and what data it contains. This is important as there may be design choices based on what information it does or does not contain.

* What is this? Is it a message
  + Is this a message – meaning is it simply a string of text?
  + Is this an object, perhaps like a JSON file (or at least a text stream that represents a JSON file so it can be constructed into that.
* What data is contained in this message? (What is the protocol?)
  + How is the message identified?
    - Is there a command or command ID that is to be interpreted?
    - Does this contain a link or a reference to the location of the Raw Data set
    - Does this contain the Meta Data for the raw data set? Or does the Meta Data need to be retrieved from some place else?

Any place where the work message is used must be figured out as to what that means, the content and structure.

### Logging and Persisting of Workflow State

This topic needs thought and to be figured out.

All workflow state, steps and actions need to be recorded and persisted. This is an audit trail and in some cases is important to be able to reproduce the same processing. It should be possible to determine the state of each workflow in its context.

# Information Gathering:

## Messages

Content and format

### Transition Service 🡪 Workflow System: POSTPROCESS.DATA\_READY

### CATALOG.DATA\_READY

Response Messages:

* CATALOG.STARTED
* CATALOG.COMPLETE

### REDUCTION.DATA\_READY

Response Messages:

* REDUCTION.STARTED
* REDUCTION.COMPLETE

### REDUCTION\_CATALOG.DATA\_READY

Response Messages:

* REDUCTION\_CATALOG.STARTED
* REDUCTION\_CATALOG.COMPLETE

## Meta Data

Where/How do we get it?

What information does it contain?

What is the algorithm for determining what WorkflowDescription to use?

## WorkflowDescription(s)

Creating some set of these is a prerequisite to having an operational system. The system should initialize itself with some set of these or be fed a set before it is asked to process a Data Set

As I see it:

* There is a pool of these
* They are not hard coded, but created and fed to the system
* This includes all the parts of the WorkflowDescription
  + Tasks
    - Actions
  + Completion/Success criteria