ARv2

Version 0.01

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 17 Nov. 2016 | 0.01 | Created | John Hetrick |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 References 1

1.4 Definitions, Acronyms, and Abbreviations 1

1.4.1 Acronyms 1

1.4.2 Terms and Definitions 1

1.5 Overview 1

2. System Overview (Black Box Context 2

2.1 Actors 2

2.2 Data Exchanges 2

3. Interfaces & Protocols 3

3.1 POSTPROCESS.DATA\_Ready 3

3.2 Other Protocol Messages 3

4. Use-Case View (High Level Functional Requirements) 3

4.1 Use Case: Reduce Data Set 4

4.2 Use Case: Add (CRUD) Workflow Description (Procedure) 4

4.3 Use Case: Rerun Existing Run Set 5

4.4 Use Case: Perform New Reduction on Existing Run Set 5

5. System Architecture (Abstract Components) 5

6. Use-Case Realizations 5

7. Design 6

7.1 WorkflowSystem 6

7.2 WorkflowEngine 6

7.3 WorkflowDescription 7

7.4 Workflow 7

7.5 Experiment 7

7.6 Actors 7

8. Design Patters, Rules, & Guideline 7

8.1 ID Strategies 8

8.1.1 Data IDs (Meta Data) 8

8.1.2 Experiment IDs 8

8.1.3 Group IDs 9

8.1.4 Sequence IDs 9

8.1.5 Workflow Description IDs 9

8.1.6 Task IDs 11

8.1.7 Workflow IDs 11

# Introduction

[The introduction of the **document subject** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this **document**. Adding text here is optional.]

## Purpose

[Specify the purpose of this document]

The purpose of this document is to ???. This is a living document which will be updated as needed to further define and clarify key decisions?

## Scope

[A brief description of the scope of this **document**; what topics it is associated with, and anything else that is affected or influenced by this document.]

This document is specific for the project. This document covers

## References

[This subsection provides a complete list of all documents referenced elsewhere in the **document**. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document. Use this to reverence web sites such as IBM infocenters as well and provide the links]

1. Reference, <link>

## Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **document**.  This information may be provided by reference to the project’s Glossary.]

This section contains the terms, acronyms and their definitions that are specific to understanding Automated Reduction Workflow.

### Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
|  |  |
|  |  |
|  |  |
|  |  |

### Terms and Definitions

| **Term** | **Definition** |
| --- | --- |
|  |  |
|  |  |

## Overview

[This subsection describes what the rest of the **document** contains and explains how the document is organized.]

<Brief overview of the Automated Reduction Workflow>

# System Overview (Black Box Context

[This subsection is the start of the primary document content. Note that bullet and number lists should be of text style Normal. Normal is single spaced] It is recommended to place the captions at the top of tables and graphics.

This section represents the context of the system under development. Of importance is the identification of the system, the actors that interact with the system, and anything that is exchanged with the system.

<context diagram>

A screenshot of a cell phone

Description automatically generated

The System is the system that manages the reduction of the data that is created as a result of neutron spallation experiment.

## Actors

This is a list of actors to the system. This includes any people, other computer systems, and processes that interact with this system.

Translation Service – The Translation Service is entity that tells the System there is a file of experiment run data from an instrument ready for processing. The following is information is exchanged by the actor:

* Message containing the location of the file to be processes

Post Processing (Reducer)

Monitor

Database (TBD)

## Data Exchanges

# Interfaces & Protocols

[This subsection is the start of the primary document content. Note that bullet and number lists should be of text style Normal. Normal is single spaced] It is recommended to place the captions at the top of tables and graphics.

<primary text>

## POSTPROCESS.DATA\_Ready

This message is generated by the Translation Service and sent to the AR Workflow System

This message can come in two formats: text string or JSON. Both formats contain the same basic information. This information is actually a path to a file on a file system. The path is not a UNC it does not contain a hostname of any kind. The assumption is that the file system is mounted to the both the system that generated the file and to the AR Workflow System. This means the path is immediately useable to get to the raw data file.

The following is the data that is contained in the path:

* Facility
  + SNS or
  + HIFR
* Instrument (ID)
* Experiment (ID)
* Run Number
* File Name

## Other Protocol Messages

Other messages contain the PATH a field for error and a field for info strings (not sure if this is one or more yet.)

* <bullet list: level 1>

<example bullet list>

* <bullet list: level 1>
  + bullet list: level 2>

# Use-Case View (High Level Functional Requirements)

[This subsection is the start of the primary document content. Note that bullet and number lists should be of text style Normal. Normal is single spaced] It is recommended to place the captions at the top of tables and graphics.

<primary text>

Notional Use Case Survey:

* Reduce Run – The Translation Service sends a message (containing a DataSet) to the Workflow System to be processed
  + The system creates determines which Procedure (recipe) to use from the MetaData
  + The system instantiates a new workflow to bind the DataSet and the Procedure(s) together
    - This needs to happen at the RunSet and a Run level
    - Consider creating a factory to do this
  + There are two different types of Recipes for the RunSet
    - One to process and stitch each run as they come in to be processed
    - One to stitch after all runs have come in and been processed
* Rerun Existing Workflow – A “user” wants to rerun an old workflow again
  + This should use the same RunSet and Runs from the original
  + This will create a new instance of a workflow. (Not overwrite the prvious)
  + This implies that there is some mechanism for a user to identify an old workflow
  + This implies there is a mechanism to tell the Workflow system to rerun the identified workflow.
    - Need to determine if there are additional runs should they be included or not? Probably yes
* Run another reduction on existing data – A run set needs to be processed again
  + This generally occurs if the Procedure has changed.

Special Circumstances

* Goal is to be able to change a Procedure, but the not the MetaData of a data set that identifies which Procedure to use.
  + Thinking of a strategy that always uses the latest “version”
  + This implies there are a set of standard Procedures that only changes by version
  + Need to determine what to do if a DataSet needs to use a truly new Procedure. I would think the MeatData would have to change in this case

## Use Case: Reduce Data Set

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

Scenarios:

* Process 1 Data Set at a time
  + First Data Set for a new Experiment and new Run Set
  + First Data Set for known Experiment, but new Run Set
  + Additional Data Set for know Experiment and known Run Set
  + [For any of the steps above] Task (step) requires multiple incoming message to be successful/complete
* Process Multiple Data Sets for Same Experiment at the same time
  + Not that the system needs to handle this, but the Use Case, and hence the system is invoked for 1 Data Set at a time. The Translation Service’s request is for one Data Set
* Process Multiple Data Sets for Multiple Experiments at the Same Time
  + Not that the system needs to handle this, but the Use Case, and hence the system is invoked for 1 Data Set at a time. The Translation Service’s request is for one Data Set

## Use Case: Add (CRUD) Workflow Description (Procedure)

This use case begins when a <user> needs to add a new Workflow Description (Procedure) to the inventory of Workflow Descriptions.

Considerations:

* Is the Workflow System going to provide a GUI for this?
* Is the new Workflow Description contained in a file (JSON or other format)?
* Need to consider the Tasks (Steps) and Actions etc. and how new ones are created/added as well (again CRUD)
  + All aspects of a Workflow Description need to be considered
* Does a new step require new code, or can it really be something the system can excute without adding new code?

Scenarios:

* <bullet list: level 1>

## Use Case: Rerun Existing Run Set

This use case begins when a <user> wants to rerun a run set that was run previously. The system will create new instances of the workflows involved but will use the exact same Workflow Descriptions that are specified in the existing Run Set and associated Runs.

Considerations:

* How does the system allow the user to identify the Run Set to be rerun?
* The Runs associated with the Run set can possibly be reduced in parallel. The Run Set Workflow Description would govern this.

Scenarios:

* <bullet list: level 1>

## Use Case: Perform New Reduction on Existing Run Set

This use case begins when a <user> wants to perform a new reduction on a set of existing data that was previously reduced. The system will create new instances of the workflows involved but will use the latest version of the Workflow Descriptions that are indicated by the MetaData.

Consideration:

* <bullet list: level 1>

Scenarios:

* <bullet list: level 1>

<example bullet list>

* <bullet list: level 1>
  + bullet list: level 2>

# System Architecture (Abstract Components)

[This subsection is the start of the primary document content. Note that bullet and number lists should be of text style Normal. Normal is single spaced] It is recommended to place the captions at the top of tables and graphics.

<primary text>

<example bullet list>

* <bullet list: level 1>
  + bullet list: level 2>

# Use-Case Realizations

[This subsection is the start of the primary document content. Note that bullet and number lists should be of text style Normal. Normal is single spaced] It is recommended to place the captions at the top of tables and graphics.

<primary text>

<example bullet list>

* <bullet list: level 1>
  + bullet list: level 2>

# Design

[This subsection is the start of the primary document content. Note that bullet and number lists should be of text style Normal. Normal is single spaced] It is recommended to place the captions at the top of tables and graphics.

The system is comprised of the following major pieces (components/packages):

* **WorkflowSystem** – This component represents the entire workflow system. This is the black box. It contains all the other pieces (components) of the systems
* **WorkflowEngine** – The workflow engine is the part of the system responsible for receiving incoming messages and invoking the correct workflow to process the message. It is also responsible for sending any actions from a workflow that are to be executed by an actor (e.g. a Reducer).
* **WorkflowDescription** – The WorkflowDescription component is the set of Tasks that have been orchestrated to processes (recipes, procedures). There are multiple WorkflowDescriptions for each representing a different set of orchestrated tasks that can be used to process data from an experiment.
* **Workflow** – Workflows are created dynamically and bind a data set (or the reference to a data set) with a WorkflowDescription. It is this Workflow that holds the state of a data set that is waiting to be, is being, or has been processed. For the AR Workflow System there are two specializations of the Workflow; one for processing a group of data sets (runs) and another for processing the data sets (runs) themselves.
* **Experiment** – The experiment is all the information about an experiment. This includes all the groups of runs (run sets), all the information and data about each specific run and any associated state information. Each workflow will be included with the experiment. Note that this Experiment is be used and referenced by systems other than the AR Workflow System. It is likely that this system will only use part of the full set of experiment information, but that is out of scope for this system.
* **Actors** – The actors package is a container for components that can act as stubs for the real systems and users of the AR workflow system. These components are mostly used for testing.

## WorkflowSystem

This component represents the entire workflow system. This is the black box. It contains all the other pieces (components) of the systems

<example bullet list>

* <bullet list: level 1>

## WorkflowEngine

The workflow engine is the part of the system responsible for receiving incoming messages and invoking the correct workflow to process the message. It is also responsible for sending any actions from a workflow that are to be executed by an actor (e.g. a Reducer).

<example bullet list>

* <bullet list: level 1>

## WorkflowDescription

The WorkflowDescription component is the set of Tasks that have been orchestrated to processes (recipes, procedures). There are multiple WorkflowDescriptions for each representing a different set of orchestrated tasks that can be used to process data from an experiment.

<example bullet list>

* <bullet list: level 1>

## Workflow

The workflow component contains the set of classes and behavior to bind data sets (or the reference to a data set) with workflow descriptions. As workflow descriptions, along with their tasks, have no state, the workflow provides the state for the processing of the data sets.

The Workflow implements the Composite pattern. This allows the workflow engine to simply invoke a generic workflow.

### Workflow

Workflows are created dynamically and bind a data set (or the reference to a data set) with a WorkflowDescription. It is this Workflow that holds the state of a data set that is waiting to be, is being, or has been processed. For the AR Workflow System there are two specializations of the Workflow; one for processing a group of data sets (runs) and another for processing the data sets (runs) themselves.

The workflow maintains most of the state while the specializations handle the specialized workflow processing. Currently, workflows sequentially walk through a set of tasks from a workflow description. However, it is the Tasks that understand how to preform the processing. As the Tasks are stateless the workflow may pass itself or part of itself to the Task so the task has context in which to operate. (Details below.)

### GroupWF

The GroupWF is a workflow handles the workflow for a group of sequences (runs). The group workflow is responsible for ensuring all sequences that belong to the group are processed appropriately. The GroupWF contains other workflows, child workflows. The child workflows are sequence workflows (SeqWF) for processing each sequence that belongs to the group.

While the workflow engine will invoke the parent workflow it is expected that the WorkflowEngine will actually be invoking the group workflow and the group workflow will invoke the appropriate child workflow.

### SeqWF

The SeqWF specializes Workflow and handles the workflow for a sequence (run).

<example bullet list>

* <bullet list: level 1>

### TaskStatus

The TaskStatus is a structure to hold the state of a task. TaskStatus are expected to be collected into a table that represents the set of status for the Tasks that belong to the workflow description.

### WorkflowRepo

The WorkflowRepo represents a workflow repository. It is the responsibility of the WorkflowRepo to return the correct workflow based on information contained in the meta data of a data set.

If necessary a workflow is created if one does not already exist.

It is anticipated that this will/could be a front end class to some sort of persistent storage of workflows. The type of persistent storage to be used has not yet been considered so the design for how this class works in that context has not yet been done.

## Experiment

The experiment is all the information about an experiment. This includes all the groups of runs (run sets), all the information and data about each specific run and any associated state information. Each workflow will be included with the experiment. Note that this Experiment is be used and referenced by systems other than the AR Workflow System. It is likely that this system will only use part of the full set of experiment information, but that is out of scope for this system.

<example bullet list>

* <bullet list: level 1>

## Actors

The actors package is a container for components that can act as stubs for the real systems and users of the AR workflow system. These components are mostly used for testing.

<example bullet list>

* <bullet list: level 1>

<example bullet list>

* <bullet list: level 1>
  + bullet list: level 2>

# Design Patterns, Rules, & Guideline

[This subsection is the start of the primary document content. Note that bullet and number lists should be of text style Normal. Normal is single spaced] It is recommended to place the captions at the top of tables and graphics.

<primary text>

## ID Strategies

There are a number of places where items in the system require an ID or a key. This section covers those cases and the strategy for creating the IDs and/or keys.

The following is a corollary document that contains some sample ID & keys used in testing: testData.xlsx.

Unless otherwise stated all of the following IDs and keys will be created, at least in part, from the values in the meta data associated with a data set. To that end a common strategy will be used where by sets of values will be concatenated and delimited by a forward slash “/”. This is consistent with the raw reference to a data set which is a location on a file system (presumably Linux hence the forward slash).

* Example:
  + instrumentID +
  + “/” + experimentID +
  + “/” + groupID
* ID/Key = “instrumentID/experimentID/groupID”

IDs are generally a single identifier for the specific piece of information such as a group or a workflow description.

Keys are most often used for searches or keys for storing the piece of information in something like a hashTable.

### Data IDs (Meta Data)

*NOTE: At this point I know these pieces of information are contained in the meta data, but I don’t know their real structure and format. All references to this data are considered abstract at this point till their real representation can be established.*

For the purposes of the prototype I have used the following convention for representing these pieces of data. There is a prefix that identifies the information type and a number (#) that identifies the ID of the information.

* Instrument ID (INST-<#>)
  + Ex: Instrument number 4: INST-4
* Experiment ID (EXP-<#>)
  + Ex: Experiment number 3054: EXP-3054
* Group ID (GRP-<#>)
  + Ex: Group number 0: GRP-0
  + Note this is the Sequence ID from the meta data
* Sequence Number (SEQ-<#>)
  + Ex: Sequence number 0: SEQ-0
* Data Type Number (DT-<#>)
  + Ex: Sequence number 0: DT-0

### Experiment IDs

The experiment ID is simply the experiment ID from the meta data.

* Experiment ID (EXP-<#>)
  + Ex: Experiment number 3054: EXP-3054

The fully qualified key would be:

* Instrument ID + Experiment ID
* Ex: INST-4/EXP-3054

### Group IDs

The group ID is simply the group ID from the meta data.

* Group ID (GRP-<#>)
  + Ex: Group number 0: GRP-0

The fully qualified key would be:

* Instrument ID + Experiment ID + Group ID
* Ex: INST-4/EXP-3054/GRP-0

### Sequence IDs

The sequence (or run) identifier is simply a number (integer) starting from zero (0). ID is simply the group ID from the meta data.

The fully qualified key would be:

* Instrument ID + Experiment ID + Group ID + Sequence Number
* Ex: INST-4/EXP-3054/GRP-0/SEQ-0

### Workflow Description IDs

Note: How to uniquely identify workflow descriptions is not yet fully figured out. Each workflow description, once used, must be retained for historical purposes. The issue is that it is yet not clear exactly how the meta data for a data set is used to identify the specific workflow description to be used. Here are some rules that have been discussed

Workflow Description for each experiment. It is expected that there will be a unique workflow for each experiment. The scientist conducting the experiment is likely to author the processing (reduction) of the data set.

Versions of Workflow Descriptions. It is likely there will be versions of the workflow descriptions for each experiment. This is because a scientist may not like the results they are currently getting and want to tweek the processing (reduction) algorithm in some small way. For instance, the formula used in a particular reduction step. Also, at some later time some scientist may want to run a group of sequences through a different workflow completely.

Default Workflow Descriptions. Currently the notion of a default workflow has been discussed where if there is a problem locating a specific workflow description for a data set, a default workflow description can/should be used be used. The discussed details are as follows. Look for a workflow using the fully qualified name of the sequence data type. If not found use the fully qualified name of the group. If not found, use the fully qualified name of the experiment.

There are actually two different types of workflows. One type for groups and one type for sequences. (See design above for details.)

#### Groups

At its core the Workflow Description ID for a group will be an identifying prefix and unique number. How to handle versions has yet to be determined.

* Group workflow description prefix: WDG-<#>
* The unique number is TBD, but current thinking is it will simply be some globally incrementing number.

The composition of the prefix is as follows:

* **WD** stands for workflow description
* **G** indicates group

Group keys are the fully qualified id of the group as follows:

* Ex: INST-4/EXP-3054/GRP-0

Group Completion Criteria are a bit different of a strategy. It is anticipated there are a limited number of group workflow descriptions – 2. These two completion criteria are as follows:

* **REDUCE&STITCH** – This is to indicate behavior to reduce the incoming data set and immediately stitch it together with the other data sets.
* **REDUCE&WAIT** – This is to indicate behavior to reduce all of the data sets before stitching them together.

#### Sequences

Each sequence does not get it own unique workflow description. The workflow description to be used on a set of sequences is expected to be derived from the data type of the data set. For this reason the workflow description ID for a group of sequences will include an identifying prefix and a unique number. How to handle versions has yet to be determined.

* Sequence workflow description prefix: WDT-<#>
* The unique number is TBD, but current thinking is it will simply be some globally incrementing number.
* Each experiment is likely to have

The composition of the prefix is as follows:

* **WD** stands for workflow description
* **T** indicates data type

Sequence keys are the fully qualified id of the group plus the data type as follows:

* Ex: INST-4/EXP-3054/GRP-0/DT-0

### Task IDs

Tasks are simple elements and don’t belong to anything until the are group in a workflow description. The identifiers reflect this with a simple prefix and a unique number. However, there are two different prefixes, one for tasks that belong to group workflow descriptions and tasks that belong to sequence workflow descriptions.

* Group Task Prefix: GTSK-<#>
* Sequence Task Prefix: TSK-<#>

### Workflow IDs

Each workflow contains a unique ID. The ID base will be created from the following pieces of data:

* Instrument ID (INST-<#>)
* Experiment ID (EXP-<#>)
* GroupID (GRP-<#>

A group workflow and a sequence workflow will vary by a prefix plus a number. The number represents which workflow it is in a possible set of workflows. The numbers start at zero (0) and increment as more workflows are added for the group or sequence. This makes the number equivalent to the index in an array as well. The prefix for each will be as follows:

* Group Workflow prefix: WFG-
* Sequence Workflow prefix: WFS-

Example:

* Group Workflow prefix: WFG-

<example bullet list>

* <bullet list: level 1>
  + bullet list: level 2>

<example numbered list>

1. <numbered list: level 1>
   1. <numbered list: level 2>

Stuff

# Thoughts, Questions, Notes

[This subsection is the start of the primary document content. Note that bullet and number lists should be of text style Normal. Normal is single spaced] It is recommended to place the captions at the top of tables and graphics.

<primary text>

## Stateless workflow

I need to read more about stateless workflows. However, a thought is that the states of a workflow could be classes that are called. The “State” class could be specialized for any activity. Events may trigger behavior in the activitity or a transition to another “State”. Some thing to think about. Maybe another solution to this AR workflow system.

## Workflow State

Currently I have some level of workflow state going to a workflow Task. In this way the Task updates the state without having to persist the state itself. This allows for the Task to be used/reused multiple times without having to create a new instant of a Task that is stateful. However, I don’t do this with the WorkflowDescription. Should I? Currently the Workflow manages the activity and behavior. Why do this? Instead would it work for the WorkflowDescription to have the behavior to process the tasks? As with Tasks the workflow state could be passed into the WorkflowDescription