SDP Science Pipeline Management
Use Case View

M. Ashdown

TABLE OF CONTENTS

1. Primary Representation

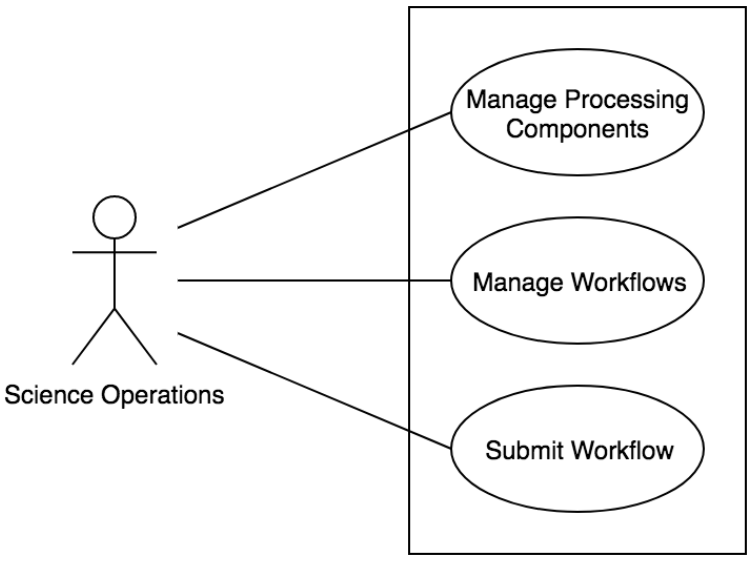

Figure 1: Science Pipeline Management Use Cases

This document considers how the science pipelines are maintained and modified, and how they are submitted to SDP. This is done from the point of view of Science Operations users interacting with the SKA system as a whole. The radio astronomy-specific parts of a science pipeline are the workflow and processing components. The workflow expresses the high-level organisation of the pipeline and the processing components perform the low-level operations. The workflow will be parameterised, so the value of its parameters must be set before it is submitted to SDP to be executed. The resulting use cases are shown in Figure 1 and analysed in more detail in Section 2.

2. Element Catalogue

2.1. Elements and Their Properties

2.1.1 Science Operations

Science Operations users will interact with the SKA system to manage the science pipelines that will be executed by the SDPs.

2.1.2 Manage Processing Components

Processing components are the low-level constituents of the science pipelines. They will be written in compiled languages. They must have unit tests to ensure that they exhibit correct behaviour, and that changes do not lead to regressions. They must also have performance tests to ensure that execution times do not change unexpectedly as a result of changes to the code. These low-level tests are an important part of the validation of the processing components, but higher-level tests as a part of workflows will also be required, since a processing component may be used by many workflows. The need for building a processing component and running extensive tests on it means that the timescale over which a new or modified processing component can be deployed to the SDPs is of the order of days.

The actions needed when managing processing components are:

- Get the source code of an existing processing component;

- Create the source code for a new processing component;
- Modify the source code of an existing processing component;
- Request the validation and, if the validation is passed, the deployment of a processing component (this could be automatically triggered by the creation or modification of a processing component);
- Get the status of a processing component (e.g. new, validated, not validated, deployed, deprecated);
- Set the status of a processing component (e.g. mark as deprecated).

2.1.3 Manage Workflows

Workflows are the high-level constituents of the science pipelines. They will be written in scripting languages. A functional validation of a workflow will consist of checking that the processing components it uses are validated, and that they are called correctly from the workflow. This will be relatively quick to run, so the timescale for validating and deploying a new or modified workflow to the SDPs is of the order of hours. A fast turn-around will be necessary if a workflow needs to be created or modified to diagnose a problem with the telescope.

The actions needed when managing workflows are:

- Get an existing workflow;
- Create a new workflow;
- Modify an existing workflow;
- Request the validation and, if the validation is passed, the deployment of a workflow (this could be automatically triggered by the creation or modification of a workflow);
- Get the status of a workflow (e.g. new, validated, not validated, deployed, deprecated);
- Set the status of a workflow (e.g. mark as deprecated).

2.1.4 Submit Workflow

Workflows are submitted to SDP via the Telescope Manager. TM does this by sending a Processing Block to SDP which specifies the workflow and the values of its parameters. Setting or changing the parameters of a workflow and validating them can be accomplished in a timescale of minutes.

There are a number of actions leading up to submitting a workflow, such as:

- Get the list of validated workflows;
- Get the specification of the parameters of a workflow (e.g. their types and range of acceptable values);
- Set the values of the parameters;
- Check the values of the parameters against the specification,

then finally:

- Submit the workflow to SDP.

2.2. Relations and Their Properties

Not applicable.

2.3. Element Interfaces

Not applicable.

2.4. Element Behavior

Figure 2 shows an activity diagram for the most important elements of these use cases: creating or modifying a processing component, creating or modifying a workflow, and submitting a processing block. These three activities are combined in one diagram because the validation of a workflow depends on its processing components being validated and deployed, and the submission of a processing block depends on its workflow being validated and deployed.

The source code for the workflows and processing components will be stored in git repositories and use semantic versioning [AD14]. Production versions of the software must be able to coexist with debugging and testing versions, so software management policies need to be developed to allow this [AD24]. Since multiple versions of the software can coexist, the version of a workflow and the versions of the processing components it uses must be traceable for debugging and testing purposes, and to be able to estimate the resources required for execution on SDP using the resource model.

Workflows and processing components may be marked as deprecated to show that they are no longer current. This provides a mechanism for managing the growth of artefact repositories containing the built software. However, it must be possible to recover deprecated software in case it is needed to reproduce previous results or for debugging and testing.

Workflows and the values of their parameters may be changed at any time up to the submission to SDP in a processing block. At this point, the SDP uses its resource model to estimate the resources required for execution, and it is added to the internal schedule. After this point the workflow and parameters cannot be changed because it may have unforeseen consequences for the schedule. To change the version of the workflow or to change its parameters, the processing block must be cancelled and replacement submitted. For this purpose it may be useful to have an atomic cancel-and-replace operation to avoid two costly recomputations of the internal schedule.

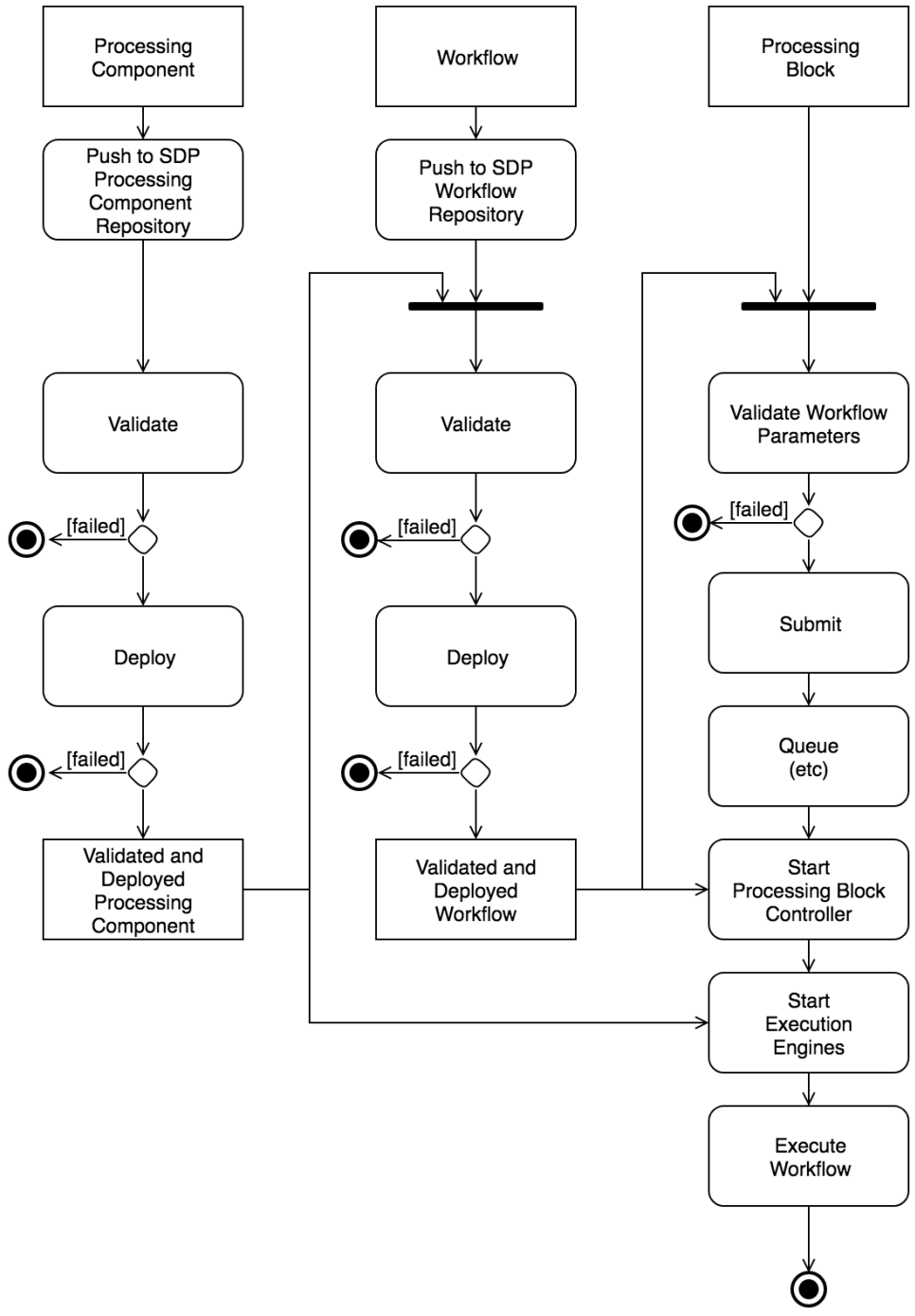

Figure 2: Activity diagram for creating or modifying a processing component (left-hand column), creating or modifying a workflow (central column) and submitting a workflow in a processing block (right-hand column).

3. Context Diagram

Not applicable.

4. Variability Guide

Not applicable.

5. Rationale

5.1 Modifiability

Requirement: SDP_REQ-809 (New pipeline workflows or algorithms)

These use cases directly address the modifiability of the science pipelines by considering the management of the workflows and processing components, including the creation of new ones, or modification of existing ones.

5.2 Validation

The nature and extent of the validation to be performed on workflows and processing components must be determined as a matter of policy.

Some of the validation should be easy to accomplish. This includes the low-level validation of the processing components, in terms of unit and performance tests, since their behaviour should be fairly deterministic. It also includes basic functional validation of workflows in terms of their interfaces. These kinds of validation steps should be required before the relevant software is permitted to be deployed on the SDP operational systems.

The more difficult questions are to do with the high-level validation of the workflows. One aspect of this is characterising the performance of a workflow, which is used to inform the SDP performance model. Another aspect is validating that the workflow produces scientifically correct results. Any requirement to do much more demanding validation of this kind before deploying the software militates against the modifiability of the science pipelines.

6. Related Views

The use cases in this document have informed the development of the SDP Software Management C&C View. The elements of the system involved in executing the science pipelines are described in the SDP Operational System C&C View and the SDP Execution Control C&C View Packet. The elements of the system involved in deployment and provisioning of software artefacts are described in the SDP Platform Services C&C View. The relationship between workflows, processing components and execution frameworks is described in the SDP System-level Module Decomposition and Dependency View. Further details on individual processing components are given in the SDP Processing Component Module View.

7. Reference Documents

7.1 Applicable Documents

The following documents are applicable to the extent stated herein. In the event of conflict between the contents of the applicable documents and this document, **the applicable documents** shall take precedence.
This list of applicable documents applies to the whole of the SDP Architecture.

[AD01] SKA-TEL-SKO-0000002 SKA1 System Baseline Design V2, Rev 03[1]

[1] This document is still a draft version and therefore not truly applicable, but will be replaced by an applicable version by System CDR.

[AD02] SKA-TEL-SKO-0000008 SKA1 Phase 1 System Requirement Specification, Rev 11

[AD03] SKA-TEL-SDP-0000033 SDP Requirements Specification and Compliance Matrix, Rev 04

[AD04] SKA-TEL-SKO-0000307 SKA1 Operational Concept Documents, Rev 02

[AD05] 000-000000-010 SKA1 Control System Guidelines, Rev 01

[AD06] 100-000000-002 SKA1 LOW SDP to CSP ICD, Rev 06

[AD07] 100-000000-025 SKA1 LOW SDP to SaDT ICD, Rev 05

[AD08] 100-000000-029 SKA1 LOW SDP to TM ICD, Rev 05

[AD09] 100-000000-033 SKA1 LOW SDP to LFAA Interface Control Document (ICD), Rev 02

[AD10] 300-000000-002 SKA1 MID SDP to CSP ICD, Rev 06

[AD11] 300-000000-025 SKA1 MID SDP to SaDT ICD, Rev 05

[AD12] 300-000000-029 SKA1 MID SDP to TM ICD, Rev 05

[AD13] SKA-TEL-SKO-0000484 SKA1 SDP to INFRA-AUS and SKA SA Interface Control Document, Rev 02

[AD14] SKA-TEL-SKO-0000661 Fundamental SKA Software and Hardware Description Language Standards

[AD15] http://www.ivoa.net/documents/TAP/

[AD16] http://www.ivoa.net/documents/latest/SIA.html

[AD17] http://www.ivoa.net/documents/DataLink/

[AD18] http://www.ivoa.net/documents/SSA/

[AD19] Memorandum of Understanding between the SKA organisation and National Radio Astronomy Observatory relating to a work package for the study and design of a new data model for the CASA software package

[AD20] MeasurementSet definition version 3.0. MSv3 team, eds. 2018. http://casacore.github.io/casacore-notes/264

[AD22] Shibboleth Authentication Service from Interenet2 https://www.internet2.edu/products-services/trust-identity/shibboleth/

[AD23] COmanage Authorization Service from Interenet2 https://www.internet2.edu/products-services/trust-identity/comanage/

[AD24] SKA-TEL-SKO-0000990 SKA Software Verification and Testing Plan

7.2 Reference Documents

None.