SHIWA Repository User Manual

This manual documents the SHIWA application repository. Sections 1-2 describe the key entities, actors and use cases, section 3-6 describe how the provided features can be accessed via the GUI.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. BACKGROUND

Researchers of all disciplines, from Life Sciences to Computational Chemistry, create and use ever-increasing amounts of complex data, and more so rely on compute-intensive modelling, simulation and analysis.

Scientific workflows have become a key paradigm for managing complex tasks and have emerged as a unifying mechanism for handling scientific data. Workflows capture the essence of the scientific process, providing a means to describe it via logical data and/ or workflows. Workflows are mapped onto concrete Distributed Computing Infrastructures (DCIs) to perform large-scale experiments.

The learning curve to use workflows, however, is demanding because workflows typically have their own user interfaces/APIs, description languages, provenance strategies, and enactment engines, which are not standard and are not interoperable. Therefore it is difficult to reuse and share workflows, this inhibits the growth in uptake and proliferation of workflows in scientific practice.

1.2. SHIWA SIMULATION PLATFORM

User communities from all around Europe use many kinds of different workflow languages. They develop workflows using one of the workflow engines. Workflow development, testing and publishing are time consuming processes and require specific expertise. These limit the number of available workflows, so it is important to share them. Workflows developed for one workflow system is normally not compatible with workflows of other workflow systems. In the past if two user communities using different workflow systems wanted to collaborate, they had to re-implement the workflows in their own workflow system. This situation can be resolved by emerging new workflow interoperability technologies provided by the SHIWA Simulation Platform.

According to the new SHIWA technologies publicly available workflows can be used by different research communities working on different workflow systems and are enabled to run on multiple distributed computing infrastructures. As a result workflow communities are not locked anymore in to their own workflow system and are able to execute workflows on several distributed computing infrastructure.

1.3. KEY COMPONENTS

The SHIWA Simulation Platform offers users production-level services supporting workflow interoperability. As part of the SHIWA Simulation Platform the SHIWA Repository facilitates publishing and sharing workflows, and the SHIWA Portal enables their actual enactment and execution in different DCIs. The simulation platform supports use cases targeting various scientific domains or subdomains will serve to drive and evaluate this platform from a user's perspective.

1.4. LINKS

The SHIWA homepage is http://www.shiwa-workflow.eu
The SHIWA Simulation platform can be found at https://ssp-test.cpc.wmin.ac.uk/liferay-portal-6.1.0/en
The SHIWA Repository can be found at http://repo-test.cpc.wmin.ac.uk/

1.5. REFRENCING WORKFLOWS AND IMPLEMENTATIONS

To reference a workflow or implementation with a URL (**U**niform **R**esource **L**ocator), copy the URL in the address bar while viewing the required workflow or implementation.
The link will be of the format:
http://shiwa-repo.cpc.wmin.ac.uk/shiwa-repo/public/edit-application.xhtml?appid=4752
or
http://shiwa-repo.cpc.wmin.ac.uk/shiwa-repo/public/edit-implementation.xhtml?impid=3208
In academic publications, please the following Harvard Style compliant method to reference workflows or implementations:
…….

1.6. HOW TO REGISTER AS A USER

Should you want to register, to enable workflow development, please send an email to shiwa-repo-admin@cpc.wmin.ac.uk.

2. ENTITY, ACTOR & USE-CASE SPECIFICATION

2.1. ENTITY DEFINITIONS

Workflow. This entity represents an abstract workflow. It describes the inputs and outputs and explains what the workflow does It also specifies sample inputs and outputs (configurations), and some further information.

Implementation. This entity represents an implementation (or concrete workflow) of a workflow. It strictly follows the input and output definitions of the abstract workflow and implements the functionality given in the workflow description. It contains or references (via e.g. URLs) the implementation description files, dependencies to run the workflow on its associated workflow engine and the workflow's graph.

Engine. This entity represents a workflow engine that is able to interpret and execute a given implementation.

User. This entity represents a repository user associated with a specific role (repo admin, workflow developer, e-scientist).

Group. This entity grants read/write/download rights to a particular workflow for a set of users (the members of the group).

Platform. This entity describes in which desktop and/or service Grid environment the implementation can be executed.

Files. This entity contains the files related to workflows and their implementations.

2.2. ACTOR DEFINITIONS

E-scientist. This actor is the consumer of the contents of the repository, i.e. workflow engines and workflows to run experiments. This actor should not register with the repository to browse and search the repository.

2.3. USE CASES

This user manual only covers the use case of the E-scientist. Use cases of all actors can be found in the administrator manual.

2.3.1. E-SCIENTIST

They are the consumers of the contents of the repository and can access the following functionality (see Figure 1).

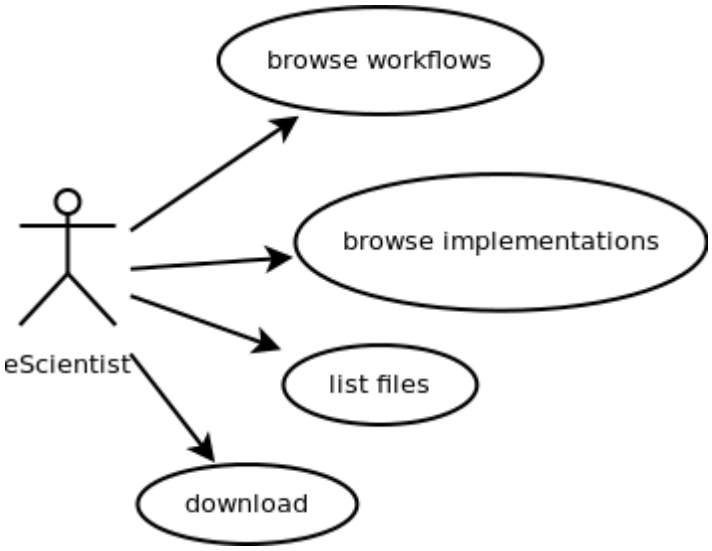


Figure 1: E-Scientist use case

Browse workflows
Browsing includes searching and listing public workflows based on their metadata.

Browse implementations
E-scientists can browse public implementations of the workflows selected by the "Browse workflows" operation.

List files
E-scientists can list files belonging to public workflows or public implementations selected by the above use cases. .

Download
Users can download workflows and their related entities (implementations, configurations and files).

3. GUI STRUCTURE

Repository features can be accessed using the main menu on the top. The following chapters go through the Workflows and Implementations tabs and describe the provided functionality. Information related to the selected tab is displayed in a table. Rows of a table can be filtered by entering text into the text field below any column title. Actions can be initiated using an “Actions” tab on the right. See illustration in Figure 2.

This is the GUI structure of the public view, which can be viewed without any login credentials.

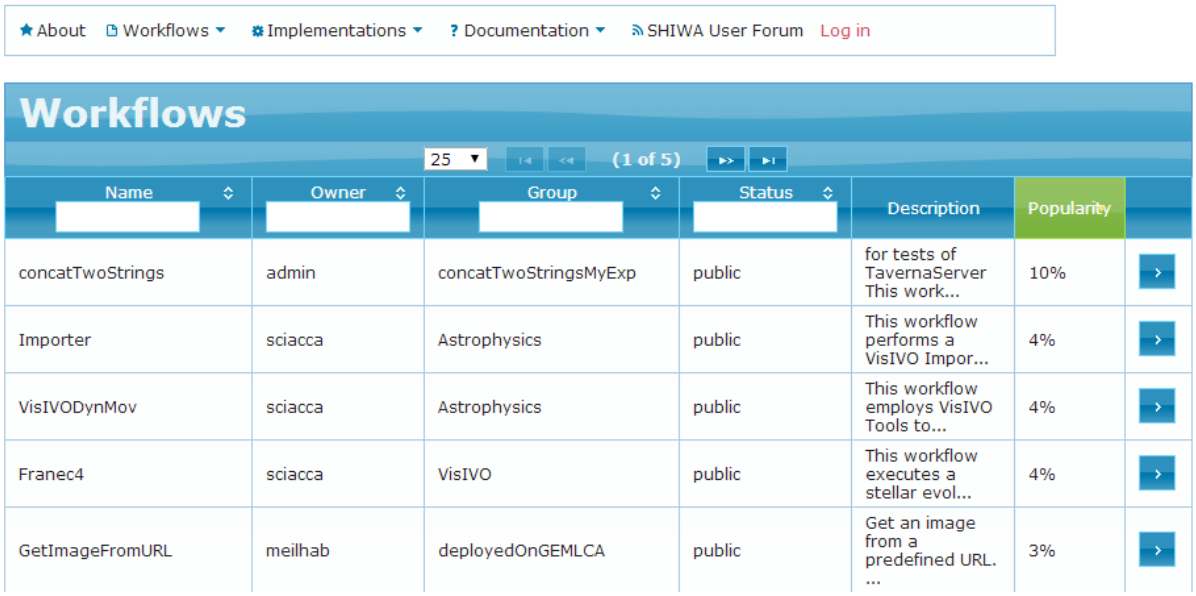


Figure 2: GUI Structure

4. WORKFLOW MANAGEMENT

4.1. LIST AND SEARCH PUBLIC WORKFLOWS

4.1.1. WORKFLOW BROWSE VIEW

Figure 3 below, displays the Workflow Browse view. This view displays basic information about public workflows and implementations in a convenient form. By clicking the Details button more information is displayed about the Workflow and all its implementations.

Inputs, Outputs and Datasets tabs can be further expanded, to display more information.

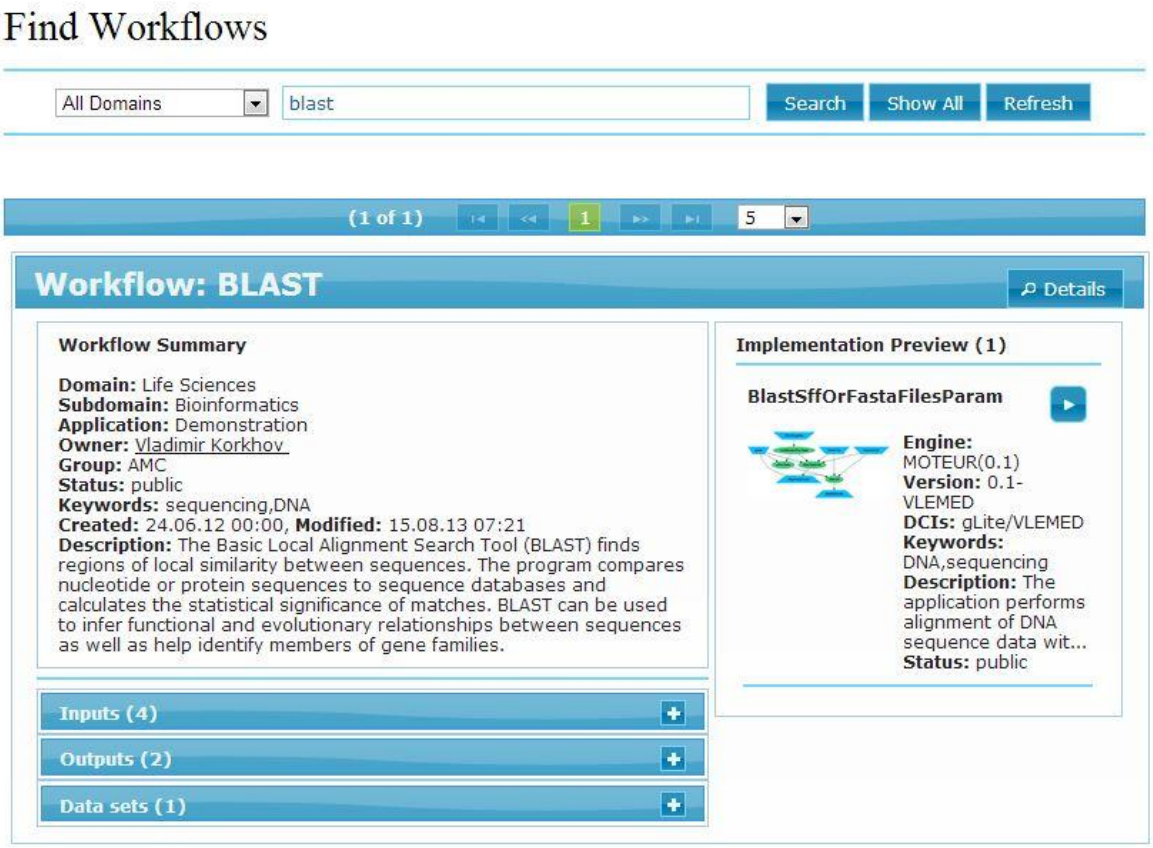


Figure 3: Workflows Browse view

Textual search in workflow or implementation records can be performed using the search box. The search operation can be restricted to application domains or subdomains.

4.1.2. WORKFLOW TABLE VIEW (FIGURE 4)

The Workflows Table view can be used to list and filter public workflows by Name, Owner, Group, Status or Popularity.

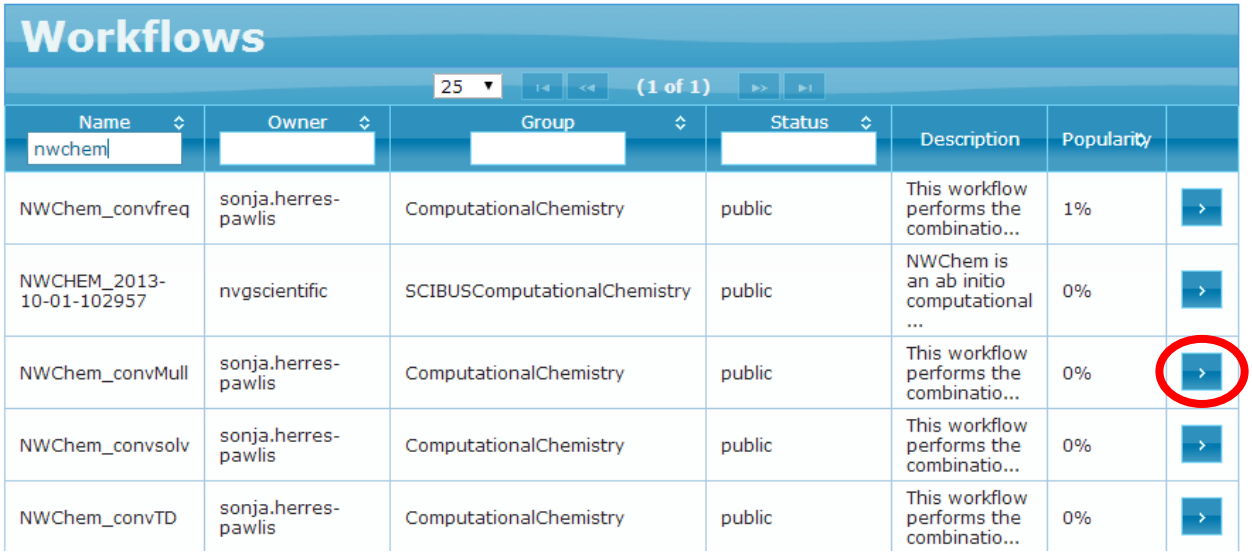


Figure 4: Workflows table view

All available information about workflows and their implementations can be viewed by selecting the workflow in this view.

To select a workflow click the Select Icon ›› of the workflow of interest. The icon has been circled in red in Figure 4.

4.1.3. WORKFLOW DETAILS (FIGURE 5)

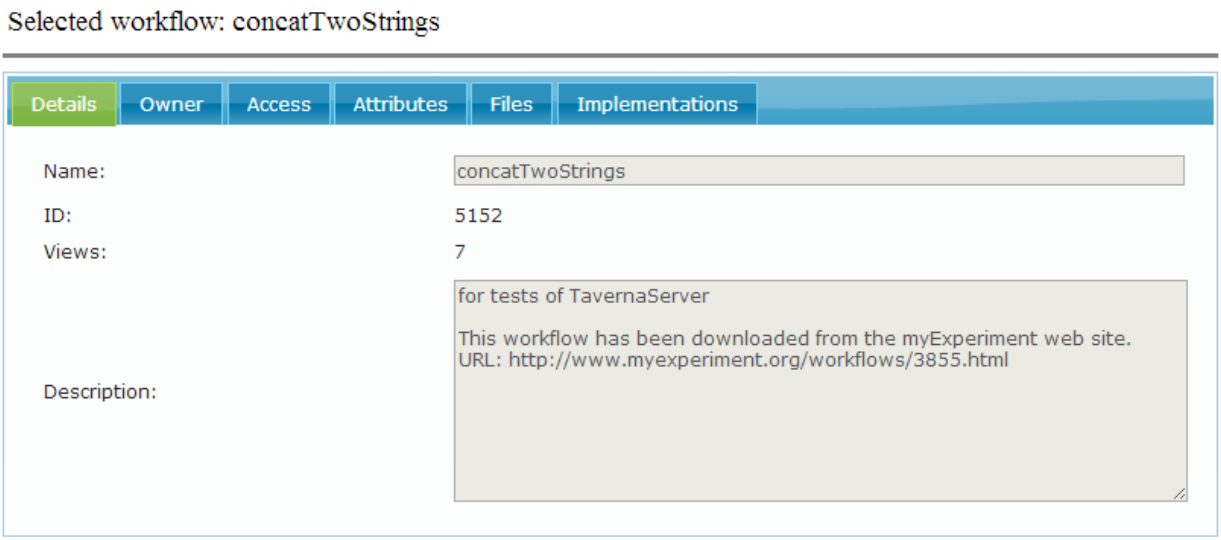
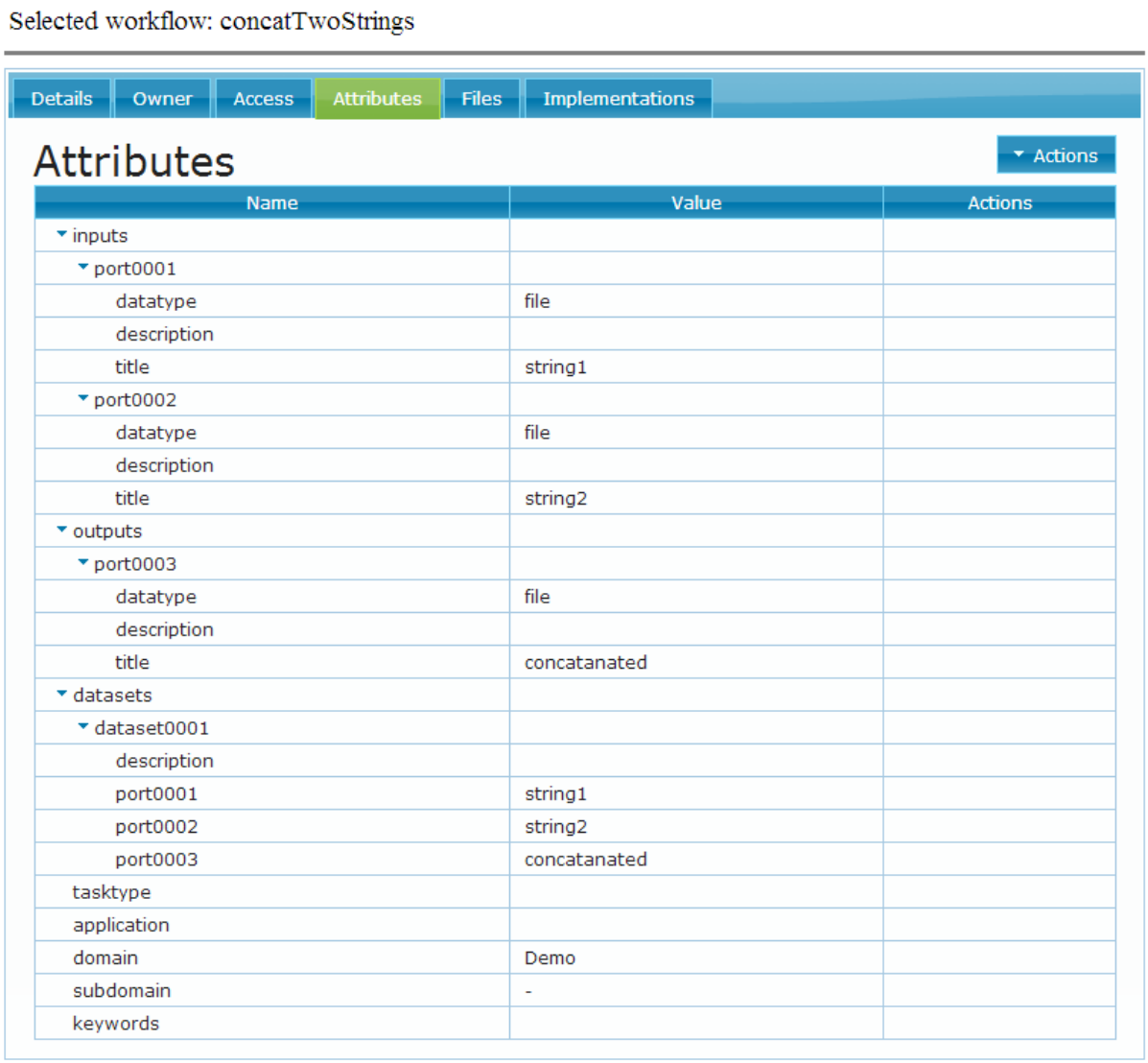


Figure 5: Workflow details page

4.1.4. WORKFLOW ATTRIBUTES (FIGURE 6)

Workflow attributes can be listed by clicking on the Attributes tab of a particular Workflow.



Name	Value	Actions
inputs		
port0001		
datatype	file	
description		
title	string1	
port0002		
datatype	file	
description		
title	string2	
outputs		
port0003		
datatype	file	
description		
title	concatanated	
datasets		
dataset0001		
description		
port0001	string1	
port0002	string2	
port0003	concatanated	
tasktype		
application		
domain	Demo	
subdomain	-	
keywords		

Figure 6: Workflow attributes page

The Expand/Collapse button in the Action control can be used to display all the attributes of the workflow.

Table Annex 1 describes the metadata structure of the attributes, and provides example values Annex 1 presents the Workflow metadata structure. These attributes allow straightforward categorisation of workflows and improve the browsing and search operations significantly. The input and output attributes with their sub-attributes define inputs and outputs of the Workflow. The dataset attribute specifies values of input parameters passed to workflow inputs, and they can also specify example outputs

4.1.5. WORKFLOW FILES & DOWNLOAD (FIGURE 7)

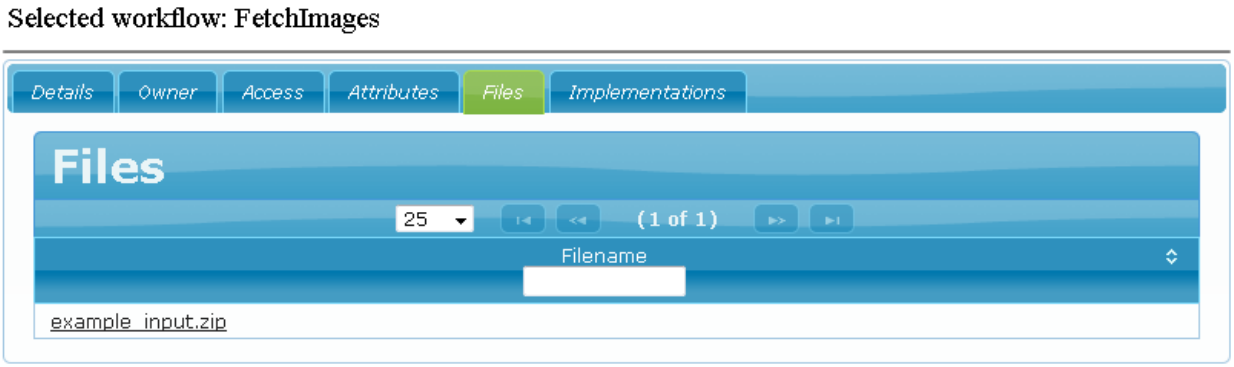


Figure 7: Workflows files page

Files associated with the abstract workflow definition can be downloaded from this page by right-clicking the files and selecting “Save as ….” or appropriate, as per your browser.

If files are not permitted to be downloaded, only their names will be displayed, but no live-links for download will be generated.

4.1.6. WORKFLOW IMPLEMENTATIONS (FIGURE 8)

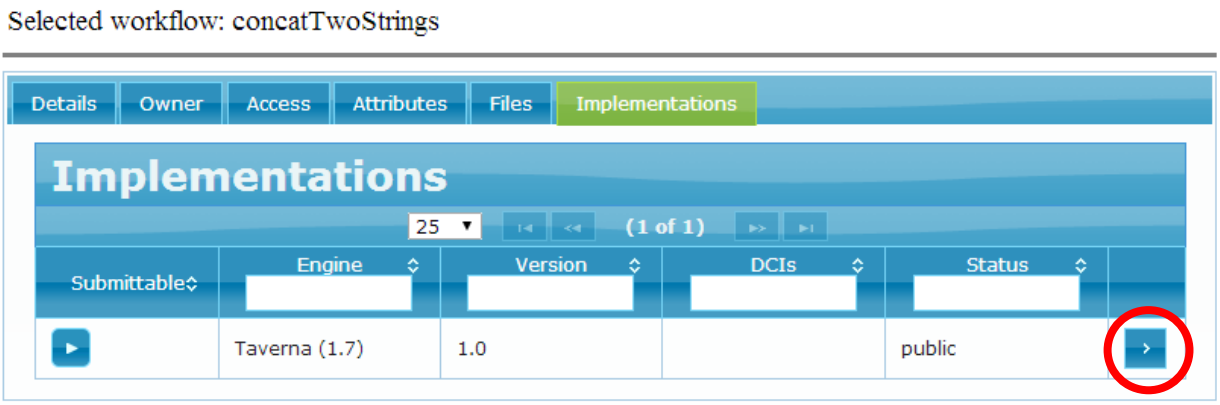


Figure 8: Workflows Implementations page

This page lists the implementations of the selected workflow.

Selecting any of the implementations directs users to the implementation details of that Implementation (see Implementations details).

To select an Implementation click the Select Icon of the workflow of interest. The icon has been circled in red in Figure 8.

5. IMPLEMENTATION MANAGEMENT

5.1. LIST AND SEARCH IMPLEMENTATIONS

5.1.1. IMPLEMENTATION BROWSE VIEW

Figure 9 below, displays the Implementation Browse view. This view displays most of the information about public implementations in a convenient form. Dependencies and Configurations can be expanded to display more information. This view also displays a 10-star review score, representing the results of the community validation. The review rating presents the average of the most recent rating given by any developer who has reviewed this implementation. Workflow owners may not rate their own implementations. Should a developer give a second review of an implementation, the previous review will be discarded.

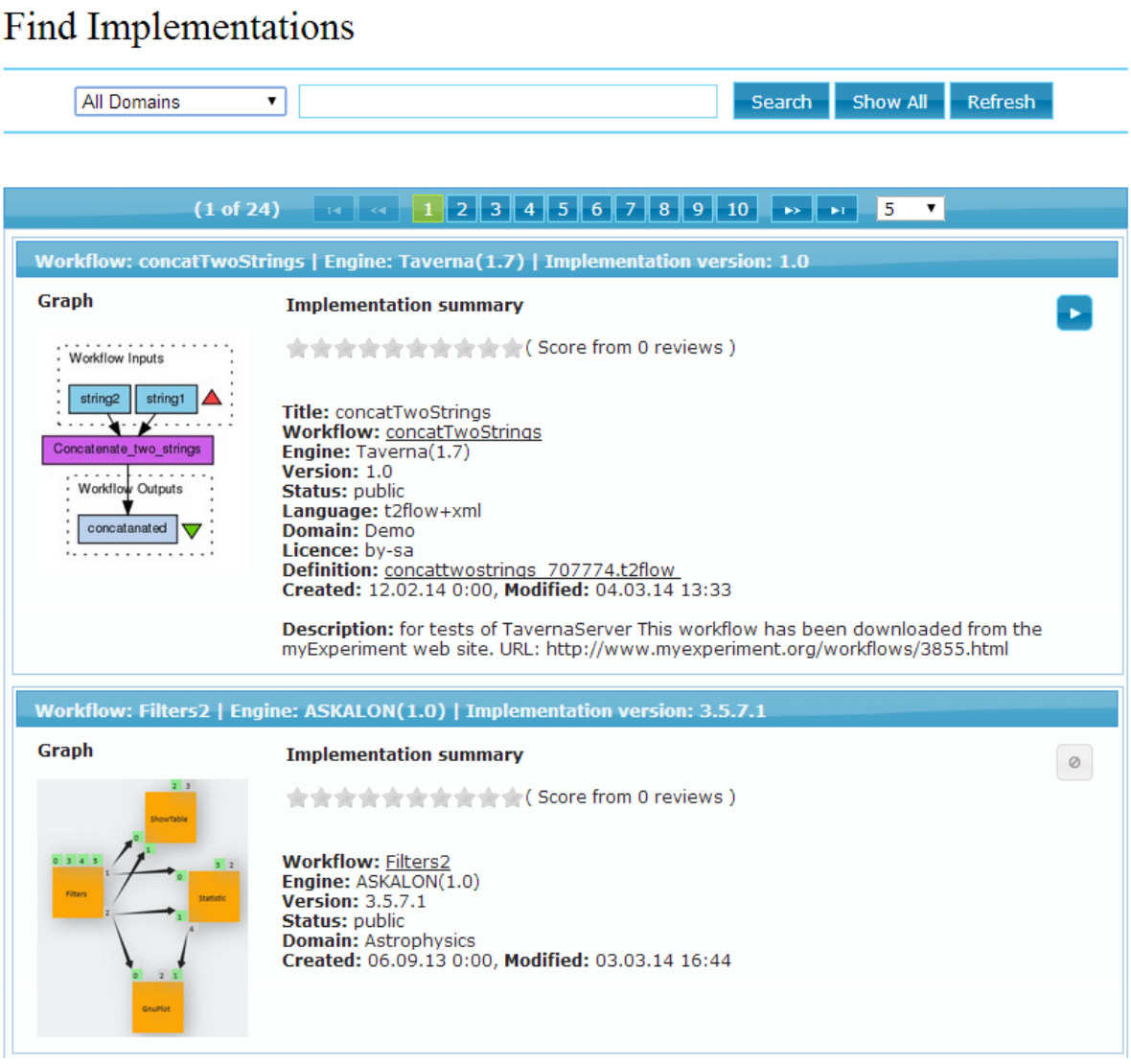


Figure 9: Browse Implementation view

Textual search in workflow implementation records can be performed using the search box. This search can be restricted to specific application domains or subdomains.

5.1.2. IMPLEMENTATION TABLE VIEW

The Implementations Table View (shown in Figure 10) can be used to list and filter public Implementations by Submitability, Workflow, Engine, Version, Status, Popularity and rating.

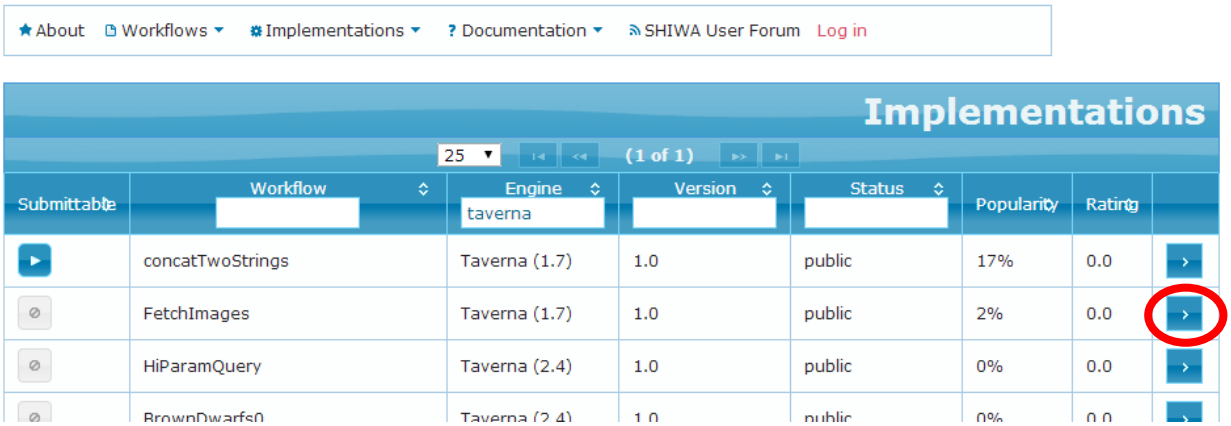


Figure 10: Implementations Table view

All available information about Implementations can be viewed by selecting the Implementation from this view.

To select an Implementation click the Select Icon ▸ of the workflow of interest. The icon has been circled in red in Figure 10.

5.1.3. IMPLEMENTATIONS DETAILS

The implementation's details page is illustrated in Figure 11.

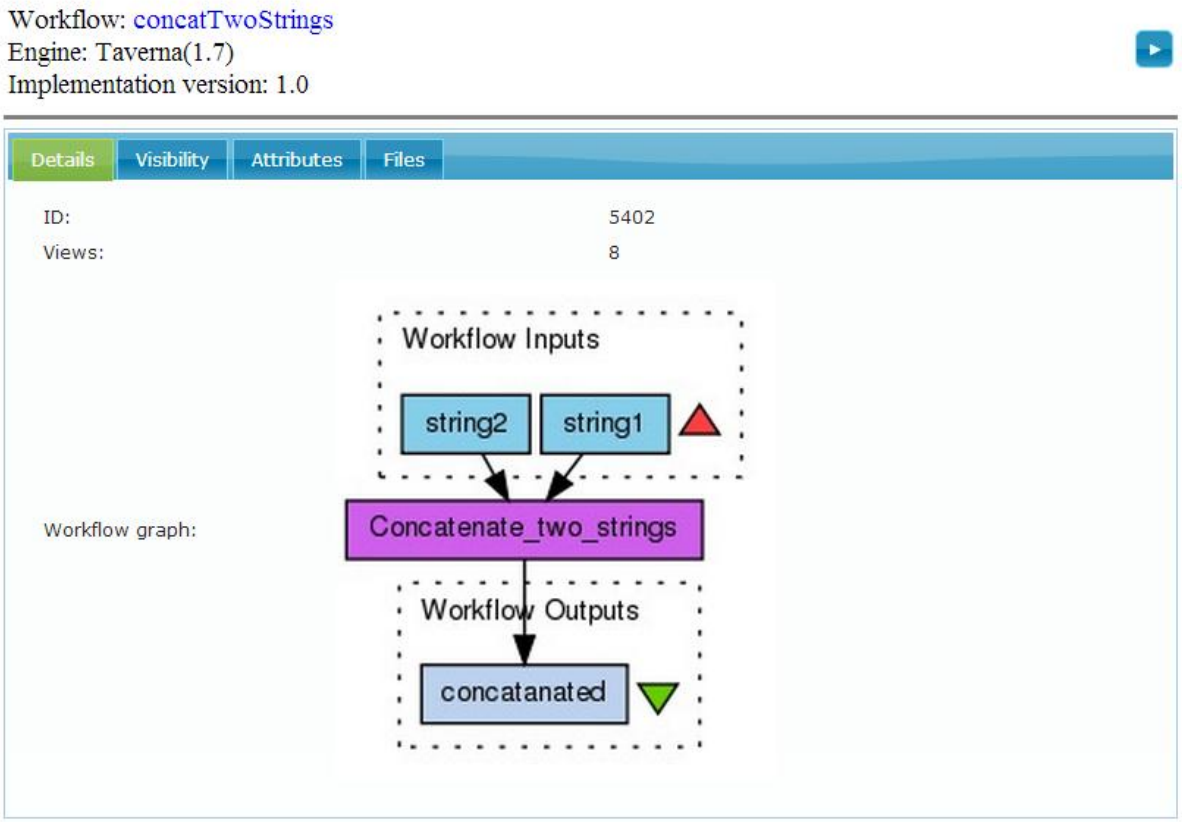


Figure 11: Implementation Details page

The *Attributes* tab can be used to open the attributes of the selected Implementation.

The *Files* tab can be used to list and download files held in the repository for the selected implementation.

5.1.4. IMPLEMENTATION ATTRIBUTES

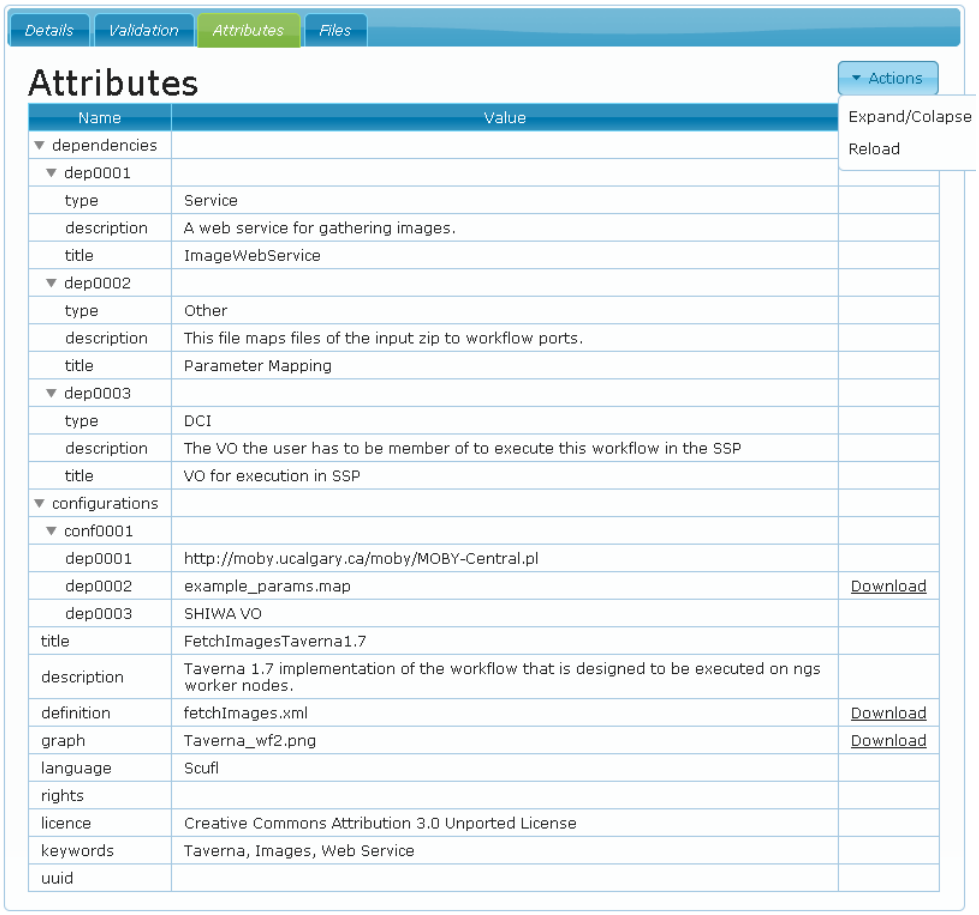
Implementation attributes can be opened by clicking on the attributes tab of a given implementation as illustrated in Figure 12.

The left column of the attributes table contains attribute names, while the right column contains attribute values.

Similarly to workflows, the metadata template is used to kelp the definition of most common attributes. The three key attributes are: definition, dependencies and configurations. The definition attribute, is the workflow definition file i.e. the executable to be interpreted by the workflow engine. The dependency attribute can be any requirement of the particular implementation. These can include for instance files, executables, libraries or VO memberships required for execution. Configuration attributes resolve these dependencies.

Table Annex 2 describes each attribute and provides example values.

Workflow: FetchImages
Engine: Taverna(1.7)
Implementation version: 1.0



Name	Value	
dependencies		
dep0001		
type	Service	
description	A web service for gathering images.	
title	ImageWebService	
dep0002		
type	Other	
description	This file maps files of the input zip to workflow ports.	
title	Parameter Mapping	
dep0003		
type	DCI	
description	The VO the user has to be member of to execute this workflow in the SSP	
title	VO for execution in SSP	
configurations		
conf0001		
dep0001	http://moby.ucalgary.ca/moby/MOBY-Central.pl	
dep0002	example_params.map	Download
dep0003	SHIWA VO	
title	FetchImagesTaverna1.7	
description	Taverna 1.7 implementation of the workflow that is designed to be executed on ngs worker nodes.	
definition	fetchImages.xml	Download
graph	Taverna_wf2.png	Download
language	Scufl	
rights		
licence	Creative Commons Attribution 3.0 Unported License	
keywords	Taverna, Images, Web Service	
uuid		

Figure 12: Implementation attribute table

The definition file can be downloaded from this page by clicking on the *download* live-link.

5.1.5. IMPLEMENTATION FILES & DOWNLOAD

As with Workflow files, these can be downloaded from the Implementation Files view, by clicking on the respective live-link (see Figure 13).

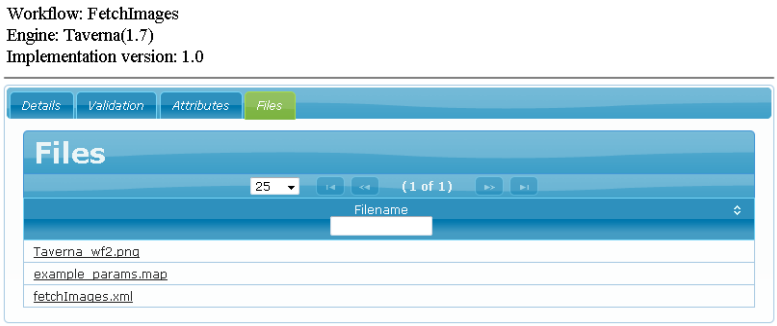


Figure 13: Implementation Files – Files tab

6. LIMITATIONS

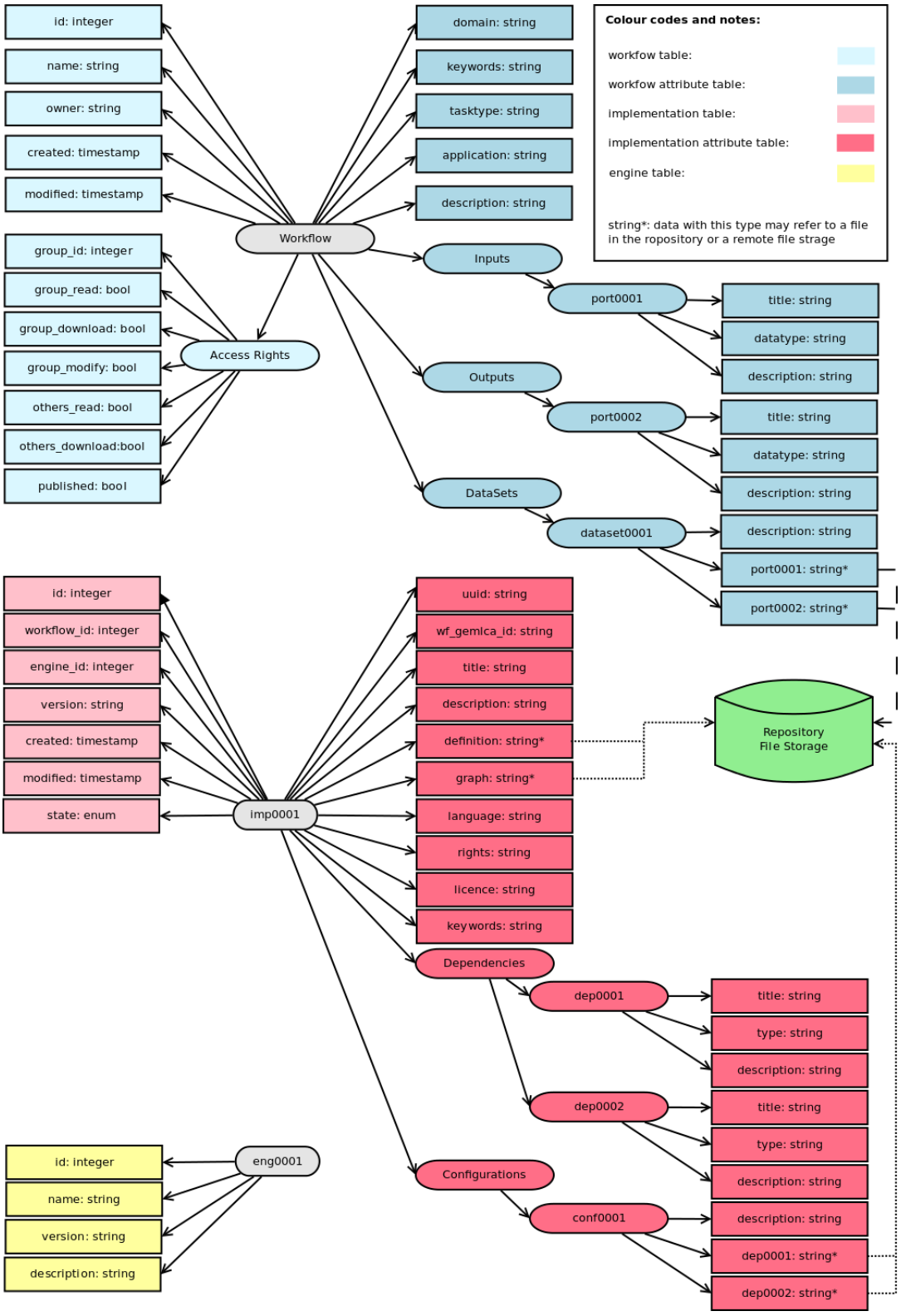
- It is not recommended to open the repository in multiple browser tabs.

Implementation metadata			Example value	Description	Table	Type	Mapping to SHIWA Desktop
id			1002	implementation identifier	implementation	int	
workflow_id			1001	identifier of the abstract workflow that the impl. implements	implementation	int	
engine_id			1005	workflow engine identifier	implementation	int	
version			1.01	implementation version	implementation	string	
created			6/15/2011 4:12	workflow creation time	implementation	timestamp	workflow->dcterms:created
modified			6/21/2011 11:24	time of last modification	implementation	timestamp	workflow->dcterms:modified
state			VALIDATED	implementation status	implementation	enum	
uuid			1234-1234-1234	uuid of implementation	imp_attr.	string	workflow->dc:identification
title			FetchImagesTaverna	title of the implementation	imp_attr.	string	workflow->dc:title
description			this implementation is ….	implementation description	imp_attr.	string	workflow->dc:description
definition			workflow.xml	workflow descriptor file	imp_attr.	string	workflow->shiwa:definition
graph			workflow.png	workflow graph screenshot	imp_attr.	string	
language			SCUFL	language of the workflow descriptor	imp_attr.	string	workflow->shiwa:language
rights			© SHIWA	copyright information	imp_attr.	string	workflow->dc:rights
licence			Demo licence	licence information	imp_attr.	string	workflow->dcterms:licence
keywords			Taverna, Images, Web Service	keywords used for searching	imp_attr.	string	
dependencies	\			List of dependencies: files needed for executing factorial.sh. It can be empty in the case of DGs.	imp_attr.		
	dep0001	\		first dependency	imp_attr.		shiwa:dependency
		title	Image Service	Title of the first dependency	imp_attr.	string	shiwa:dependency->dc:title
		description	A web service for gathering images.	Description of first dependency	imp_attr.	string	shiwa:dependency->dc:description
	dep0002	\			imp_attr.		shiwa:dependency
		title	Parameter Mapping	Title of the second dependency	imp_attr.	string	shiwa:dependency->dc:title
		description	This file maps files of the input zip to workflow ports.	Description of second dependency	imp_attr.	string	shiwa:dependency->dc:description
configurations	\			List of dependency configurations. A configuration resolves all dependencies of the executable. It can be empty if no dependencies.	imp_attr.		
	conf0001	\		first configuration	imp_attr.		
		description	This configuration...	A description of the configuration	imp_attr.	string	shiwa:configuration->de:description
		dep0001	http://moby.ucalgary.ca/…	to resolve dep0001	imp_attr.	string	shiwa:dependencyref->rdf:value
		dep0002	example_params.map	to resolve dep0002	imp_attr.	string	shiwa:dependencyref->rdf:value

Table Annex 1: Workflow metadata attributes

Workflow metadata			Example value	Description	Table	Type	Mapping to SHIWA Desktop
id			1001	workflow identifier	workflow	int	
name			Factorial	workflow name	workflow	string	
owner_id			1008	workflow owner id	workflow	int	
group_id			exampleGroup	user group for defining access rights	workflow	int	
group_read			TRUE	whether group members can see wf/impl. data	workflow	bool	
group_download			TRUE	whether group members can download wf/impl. files	workflow	bool	
group_modify			TRUE	whether group members can modify wf/impl. data and upload files	workflow	bool	
others_read			TRUE	whether registered users can see wf/impl. data	workflow	bool	
others_download			TRUE	whether registered users can download wf/impl. files	workflow	bool	
published			TRUE	whether unregistered users can see wf/impl. data and download files	workflow	bool	
created			6/5/2011 13:12	workflow creation time	workflow	timestamp	
modified			6/7/2011 16:59	time of last modification	workflow	timestamp	
application			GATE	name of the application which the wf is part of	workflow_attr.	string	workflow->shiwa:application
description			This workflow …	workflow description	workflow_attr.	string	
domain			Mathematics	scientific domain	workflow_attr.	string	workflow->shiwa:domain
keywords			factorial, integer	workflow keywords	workflow_attr.	string	
tasktype			demo	type of task the workflow represents	workflow_attr.	string	workflow->shiwa:tasktype
inputs	\			list of workflow inputs			
	port0001	\		first input port			shiwa:inport
		title	PositiveInteger	name of the port	workflow_attr.	string	shiwa:inport->dc:title
		datatype	file	data type of port	workflow_attr.	string	shiwa:inport->rdf:datatype
		description	this file contains an integer	port description	workflow_attr.	string	shiwa:inport->dc:description
outputs	\			list of output ports			
	file0002	\		first output port			shiwa-outport
		title	Factorial	name of the port	workflow_attr.	string	shiwa:outport->dc:title
		datatype	file	data type of port	workflow_attr.	string	shiwa:outport->rdf:datatype
		description	this file contains the factorial of the input integer	port description	workflow_attr.	string	shiwa:outport->dc:description
datasets	\			List of input/output configurations			
	dataset0001	\		First configuration			
		description	This dataset...	An examle dataset	workflow_attr.	string	shiwa:dataset->de:description
		port0001	input.dat	example value for port0001	workflow_attr.	string	shiwa:portref->rdf:value
		port0002	output.dat	example value for port0002	workflow_attr.	string	shiwa:portref->rdf:value

Table Annex 2: Implementation metadata attributes



Annex Figure 1: SHIWA Repository metadata graph