

Asynchronous services in Taverna

Stian Soiland-Reyes and Christian Brenninkmeijer
University of Manchester

<http://orcid.org/0000-0001-9842-9718>

<http://orcid.org/0000-0002-2937-7819>

Bonn University, 2014-09-01 / 2014-09-03

<http://www.taverna.org.uk/>

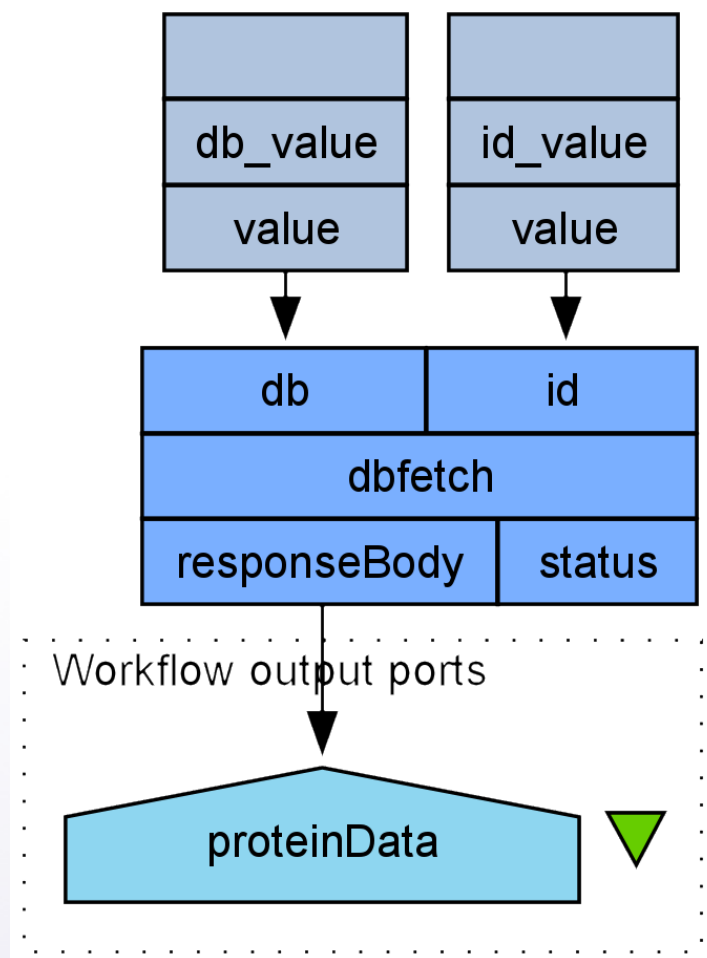


*This work is licensed under a
[Creative Commons Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/)*



Synchronous web services

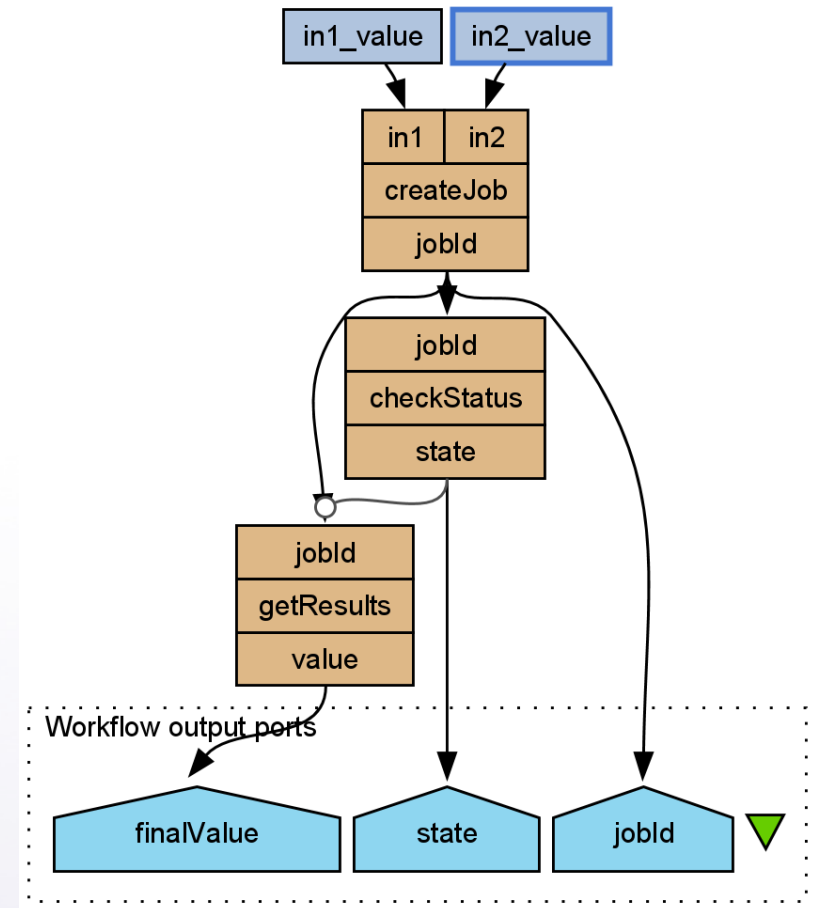
- Most web services (REST and SOAP) are **synchronous**
 - No response until the processing/results are ready
- Problem:
 - Long-running calculations (e.g. sequence alignment) can time out on the network
 - Can't tell if the service is taking a long time to process, or is simply failing to respond





Asynchronous services

- **Asynchronous** services on the other hand works by multiple calls in a fixed pattern.
- Typical pattern:
 - *createJob* takes any inputs, and returns some kind of *jobId* – e.g. 001572
 - *checkStatus* return the current *state* of the job – e.g. RUNNING
 - *getResults* retrieve the outputs for the job





How to recognize an asynchronous service?

- Although this is a common pattern, services vary in what they call each step and state. Typical names:
 - createJob, submit, run, runAsync, launch, sequenceAlignment
 - Returning *jobId*, *job*, *id*, *run*
 - checkStatus, job_state, check_job, isFinished
 - Might return more than just the status
 - getResults, retrieveOutput, jobOut
 - Sometimes merged with *checkStatus*
 - RUNNING, active, QUEUED, Finished, complete, ready

https://www.ebi.ac.uk/biocompare/services/3679/service_endpoint

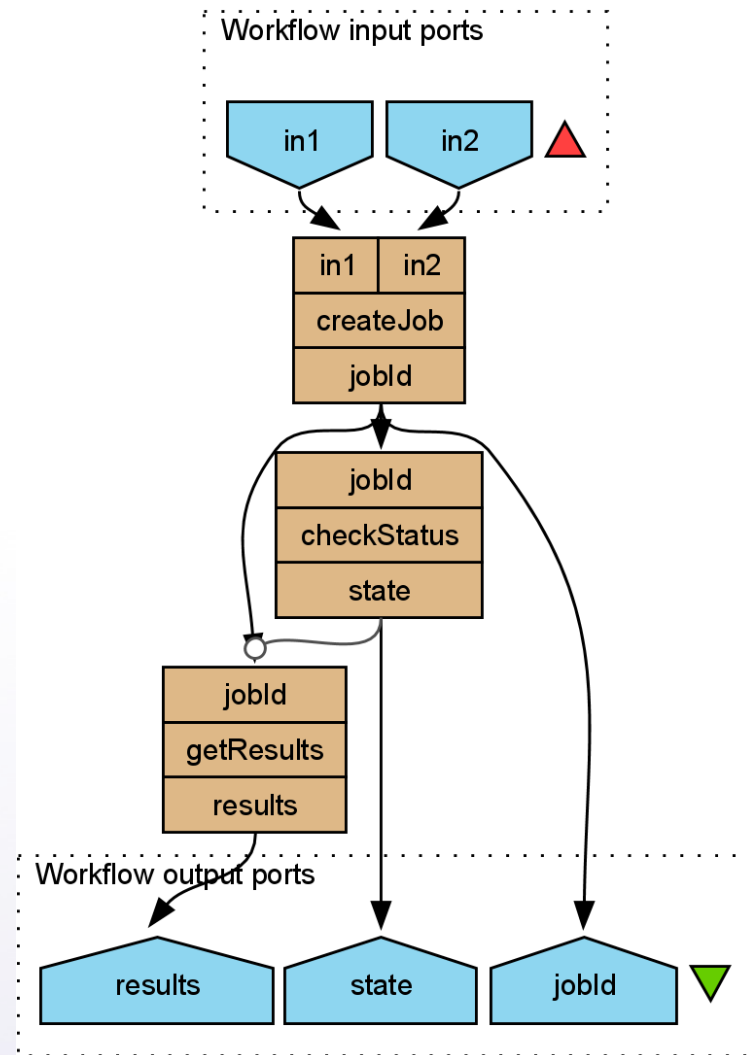
The screenshot displays the InterProScan 5 (SOAP) service interface. At the top, it identifies the service as 'InterProScan 5 (SOAP)' with a 'SOAP' button. Below this, it lists the 'Specified Name (from WSDL): JDispatcherService' and alternative names 'InterProScan (SOAP)' and 'InterProScan'. A 'Categories' section includes 'Domains', 'Motifs', 'Protein Motifs', 'Sequence Analysis', 'Protein Sequence Analysis', and 'Function Prediction'. The main content area is titled 'Overview' and 'Soap Services (6)', with sub-sections for 'Examples', 'Monitoring', and 'History'. A yellow banner states: 'These are the SOAP operations available for the SOAP variant of this service. Click on each one to get more information and to...' Below this is a 'Quick Browse' section with links to 'getParameterDetails', 'getParameters', 'getResult', 'getResultTypes', 'getStatus', and 'run'. The interface then lists several SOAP operations, each with a brief description and 'Tags on this SOAP Operation: none':

- getParameterDetails**: 1 input | 1 output. Part of Service: InterProScan 5 (SOAP). Description: 'Get some details about a parameter (e.g. name, description, values, etc.)'.
- getParameters**: 1 input | 1 output. Part of Service: InterProScan 5 (SOAP). Description: 'List the names of the parameters available before submission'.
- getResult**: 1 input | 1 output. Part of Service: InterProScan 5 (SOAP). Description: 'Get a job result formatted using a particular renderer'.
- getResultTypes**: 1 input | 1 output. Part of Service: InterProScan 5 (SOAP). Description: 'Get the list of renderers available to output a job result (i.e. the list of available output types)'.
- getStatus**: 1 input | 1 output. Part of Service: InterProScan 5 (SOAP). Description: 'Get the status of a submitted job'.
- run**: 1 input | 1 output.



Asynchronous service pattern in Taverna

- A **control link** ensures *getResults* does not run until *checkStatus* is finished
- **Looping** on *checkStatus* to repeat until status is no longer RUNNING
- Covering both FINISHED and FAILED
- Might be a nested workflow
- Wrapped as a **nested workflow** or **component** to appear as a single service within mother workflow





Exercises

- “Looping” Exercise (in Advanced Taverna) details how to configure **looping** in a workflow using the EBI *InterproScan* protein signature recognition service.
- “Control Links” Exercise details how to add **control links** to ensure results are not fetched before the job is complete.