

CP5046 : Project

SAL-T : Project Overview Statement

Jochen Braun, Andreas Knirsch, Andreas Seemann
September 3, 2008

This document is intended to give the reader a brief overview to SAL-T. The structure is derived from [Wys06]. The project itself is part of subject CP5046 in performed in semester 2-2008 at the James Cook University (JCU), Townsville.

The accompanying lecture is held by Prof. Ian Atkinson. The project is supervised by Chris Christensen.

1 Problem

To support large sensor networks the Sensor Abstraction Layer (SAL) can be used to simplify sensor network access, control and management. It reduces the effort to integrate different sensors by abstracting the underlying technology. Once SAL has identified the sensors data can be collected by SAL. The problem is now how can a huge amount of data be transmitted to various consumers. One of the consumers would be the SRB, a data repository where all this sensor data can be stored and easily accessed.

1.1 Scope

The general aim of this project is to connect various sensors to various consumers. SAL is already established to gather data from different sensor types and make them accessible in an unified way. The task is to set up a middleware between SAL and any possible consumers. Data that is collected from any sensor has to be published in this middleware. The SRB will be a consumer that retrieves the data and stores it for later processing. DataTurbine will be the middleware of our choice, because it is the favorite product of our client.

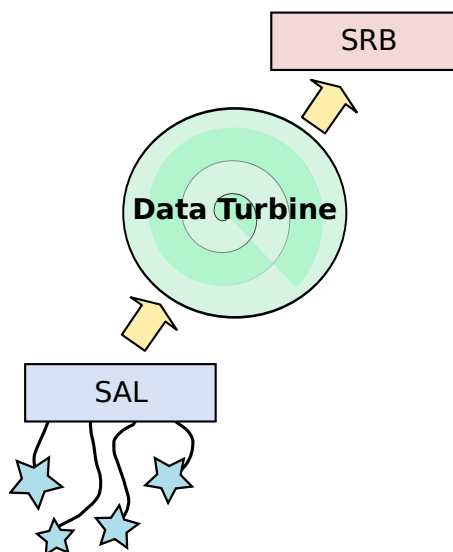


Figure 1: System Overview

SAL The Sensor Abstraction Layer (SAL) provides a coherent and consistent programming interface to access sensors and instruments regardless of their types and technology. SAL documentation is available at [Gig].

SRB is an application which provides a logical interface to heterogeneous storage devices. The most defining feature of SRB is the MCAT (Metadata Catalogue) which allows complex metadata descriptions to be attached to files stored by SRB. The most common usage of SRB is as a Distributed Logical File System (a synergy of database system concepts and file systems concepts) that provides a powerful solution to manage multi-organizational file system namespaces. [Cen]

Data Turbine is a robust open-source streaming data middleware system that satisfies the core requirements for sensor-based environmental observing systems. It facilitates the development of complex distributed streaming data applications and provides the foundation for reliable data acquisition and instrument management services. [Ini]

1.1.1 Complexity

The project's complexity is high, because of the following reasons:

- two different technologies have to be integrated, it is not just one system to deal with.
- we have two different clients with different needs.
- at the beginning of the project we have no expertise in the existing systems.
- the used systems are research projects and still under development.
- the finished system will face high data throughputs.
- our implementation will be difficult to test in the integrated system.

2 Goal

The general goals our two clients have given us are as follows:

Gilles: Implement a glue between SAL and DataTurbine to send all data of the sensors known by SAL. The sample rate of each different sensor has to become configurable.

Nigel: Get the sensor data into SRB and register it in the repository.

2.1 Functional Requirements

1. The performance and configuration of Data Turbine shall be quickly evaluated.
2. The data related to each sensor connected to SAL shall be read and sent to Data Turbine. A sensor can be either one of the following list:
 - video
 - 1-wire
 - snmp
 - os data
3. The connection to SAL shall use RMI.

4. The data sent to DataTurbine? shall include metadata, which is in first place the timestamp and the sensor-id.
5. Sampling rate for sensor data reading shall be configurable depending on the sensor type.
6. (optional) The connection to SAL shall use local objects/direct communication.
7. Sensor data shall be dumped into the SRB
8. The format of the sensor data stored into SRB shall be specified; it shall be optimized for data retrieval
9. The metadata which comes with the sensor data itself shall be stored into SRB
10. The metadata format shall be extensible
11. To access SRB the commons-vfs shall be used

2.2 Non-Functional Requirements

1. The project results shall be available under a open source license.
2. Results of some performance tests shall be provided

3 Objectives

This section outlines the tasks the authors currently have planned to perform to adequate address the problem described in section 1 and fulfil the goal pictured in section 2.

3.1 Project Management Approach

There are some facts, which make the project hard to manage: The project members do not have much domain knowledge in the projects context, they do not have expertise in the software products to use for solving the underlying problem and the projects content was only roughly contoured during the first interviews with the clients.

As consequence an *Adaptive Project Management* approach will be used to perform the project. Fortunately this is supported by the on-campus availability of both clients. It is planed to make use of this circumstance by involving the clients into the development process to ensure the realisation of the system as close to the clients needs as possible.

3.2 Design

As already mentioned the projects nature is mainly a software implementation targeting on the integration of independent already existing software products. This includes that human user interaction is limited. Therefore not much weight lies on a high sophisticated HMI (Human Machine Interface). But in contrast the API (Application Programmers Interface) of the three involved software products should earn reasonable attention. Their task can be summarized as follows:

SAL Provider of data gathered by connected sensor networks.

Data Turbine Middleware which takes the part of a transport layer.

SRB Consumer of the transmitted data to store for further usage.

A coarse grained sketch is presented in Figure 2 to visualize the dependencies between the software products. The yellow arrows in between represent the necessary glue to connect the different worlds. The development of this glue is the main objective of this project.

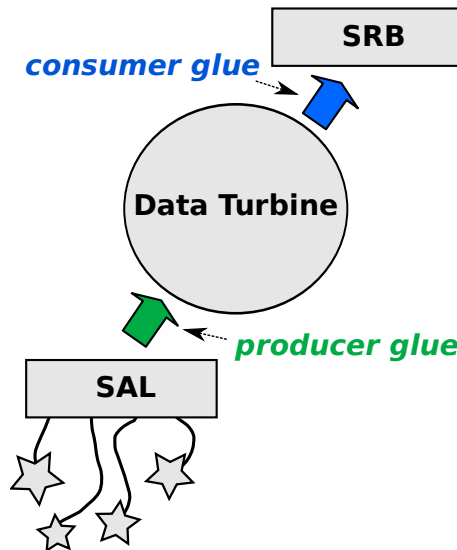


Figure 2: The glue to connect producer and consumer via a middleware.

To breakdown the overall project into subparts to reduce the scope for certain development steps, the authors choosed the following structure:

Infrastructure Tasks to be performed for both producer and consumer, like the setup of a development environment including a configuration managment, installation of the middleware Data Turbine and the establishment of a test suite.

Producer Glue This part addresses all necessary efforts to send data from SAL into the middleware. This includes the installation of SAL and the development of a suitable wrapper or client which fetches available data from SAL as well as routines to promote those data into the middleware. Therefore ways how to register and unregister certain data formats like streaming media have to be determined.

Consumer Glue This part addresses all necessary efforts to fetch data by a consumer out of the middleware to process or store for further purpose. As well ways how to subscribe and unsubscribe for certain data sources and certain data formats like streaming media have to be determined.

3.3 Implementation

In respect to the choosen adaptive project managment approach (see subsection 3.1) the realisation of the described system will be performed using iterative cycles.

3.3.1 Infrastructure

As development environment the *Eclipse IDE* is selected [Fou]. It supports the implementation of the necessary JAVA code, source code management with *Subversion* as well as the preparation of project documents with \LaTeX . Therefore *Eclipse IDE* forms a single IDE for all the projects development tasks.

For the source code management, which also includes the documentation to the project *Subversion* is used. As service provider for hosting the related project management server

services *Google Code* was determined to fit the project needs best (see [BKS]). It provides a *Subversion* repository, a project Wiki as well as a issue tracking system to be used for managing software development and maintenance tasks while promising to take care of data security in terms of a sufficient backup concept. Further all project members have access to the project artifacts independent of their location¹ as long they have access to the internet. Also this ensures the outcome of the one-semester based project is accessible by a broader community for usage and improvement.

For the installation of the middleware Data Turbine some adequate PC hardware is provided by the client. The necessary steps to setup an operational system will be recorded as they are part of the requirements (see subsection 2.1).

To support the agile idea of co-location of the project members, the project supervisor will provide an adequate team space. The establishment of the described infrastructure conditions any further proceeding and will be issued within the first project cycle.

3.3.2 Producer Glue

An operative SAL installation is essential for the producer part of the project. The client already has provided sufficient information to establish a SAL server which can be accessed via a SAL client.

According to the clients requirements (see subsection 2.1) a new instance of a SAL client will be realised, which polls the sensors attached to a SAL server using RMI (Remote Method Invocation). The interval of the polling depends on the type of the sensor and has to be configurable to respect the different needs for different types of data (discrete measurements of physical attributes, video streaming, etc.).

The implementation will make use of JAVA, due to the fact both SAL and Data Turbine are providing JAVA APIs, only.

3.3.3 Consumer Glue

An operative SRB (Storage Resource Broker) is essential for the consumer part of the project. The client has promised to grant access to an SRB for development issues.

Further details related to used technologies for implementation have to be determined, following one Poppendiek's Lean Software Management maxim "*Decide as late as possible*" (see [PP03]).

3.4 Test

The authors amplify the need for testing by setting up appropriate test routines using the JUnit test framework at the beginning of each cycle. This helps to reduce problems caused by ambiguous understanding of the related requirements as well as to declare a unit of software as finished successfully before having the unit successful tested.

Further, unit test procedures might improve maintainability due to reusability after slight modification of parts of the system in future as well as they help future developers to understand certain parts of the system.

4 Success Criteria

This section describes how the authors have planned to validate the projects outcome to make sure to meet the clients needs. This corresponds to the clients acceptance criteria.

¹This is essential for the off-campus living project members.

For each of the given criteria a brief explanation in prose is given followed up by a short summary categorised in pre and post-condition.

4.1 Infrastructure Related Criteria

4.1.1 Performance and Configuration of Data Turbine

There is comprehensive documentation available to install and configure Data Turbine either in form of a script or step-by-step instruction in such a way, that it is operational. Instructions are also provided to measure performance of Data Turbine and monitor processed data.

Pre-Condition A standard linux distribution installed on a x86 32-Bit hardware.

Post-Condition A operational Data Turbine installation.

4.2 Producer Glue Related Criteria

4.2.1 Communication with SAL via RMI

There is a software implementation available, which is able to connect to a remote running SAL server agent.

Pre-Condition A operational SAL server agent accessible via RMI.

Post-Condition SAL-T displays connected sensors to the remote SAL server agent.

4.2.2 Sensor data is read and sent to Data Turbine (DT)

There is a software implementation available which is able to connect to a running SAL server agent. For each sensor connected to the SAL server agent the software implementation polls in configurable intervals for the data provided by the sensor. The interval is configurable individually for each sensor type in milliseconds before execution time. The supported sensor types are listed below:

- video
- 1-wire
- snmp
- os data

The fetched data is send to the DT, which can be visualised by a monitoring tool attached to DT.

Pre-Condition A operational DT installation and an operational SAL server agent.

Post-Condition Data for each connected/specified sensor is send to DT in defined intervals.

4.3 Sensor data is enriched with metadata when sent to Data Turbine

There is a software implementation available which is able to correlate data fetched from sensors with metadata. This metadata consists of the listed below.

- sensor identifier
- timestamp

Pre-Condition A operational DT installation and an operational SAL server agent.

Post-Condition Data for each connected/specified sensor is sent to DT with sensor identifier and timestamp.

4.3.1 Communication with SAL via local objects/direct communication

This is an optional success criteria.

There is a software implementation available, which is able to be connected to a local running SAL server agent. The connection is realised in such a way that the software implementation is getting part of SAL. This implies there is only one java class needed to startup SAL together with SAL-T.

Pre-Condition A operational SAL server agent.

Post-Condition SAL-T displays connected sensors and their data when started with SAL.

4.4 Consumer Glue Related Criteria

4.4.1 Format of sensor data to be stored in SRB is specified

There is a documentation available describing the format to be used for storing sensor data within a SRB.

4.4.2 Sensor data is dumped into a SRB

There is a software implementation available which is able to receive sensor data from the Data Turbine and store it within a SRB using the determined format of subsubsection 4.4.1. To abstract from the SRB in order to provide a higher degree of flexibility Commons VFS with GRID enhancements is used. The incoming data can be visualised with already available SRB browser (see [HPR]).

Pre-Condition A operational DT installation, an operational SAL server agent and an accessible SRB.

Post-Condition An SRB browser is able to visualise sensor data.

4.4.3 Metadata which comes with the sensor data is dumped into a SRB

There is a software implementation available which is able to receive metadata together with related sensor data from the Data Turbin and store it within a SRB using the determined format of subsubsection 4.4.1.

Pre-Condition A operational DT installation, an operational SAL server agent and an accessible SRB.

Post-Condition An SRB browser is able to visualise sensor data related with metadata.

5 Assumptions

This section is intended to provide the authors' assumptions regarding the project performance influenced by risks and hindrances.

5.1 Risks

Title	<i>Requirement Understanding</i>
Severity	high
Description	Requirements are only partly known at project start.
Mitigation	Fixate high level clients requirements at early stage. Meet regularly with client and professionals to evaluate requirements understanding.

Title	<i>User impact understanding</i>
Severity	high
Description	Difficulties in understanding impact on users. Project team has no contact to real end users.
Mitigation	Frequent meetings with client to enable lifelike development cases.

Title	<i>Project Scope</i>
Severity	medium
Description	Size of project task exceeds timeframe.
Mitigation	Talking to the client and professionals in the field (Gilles, Nigel) to get input and feedback about used technology and approaches. Splitting project into several cycles to be able to deliver a working prototypical part if needed [Wys06].

Title	<i>System familiarity</i>
Severity	medium
Description	Project members lack insight into field of profession and technologies.
Mitigation	Get in touch with professionals of the field (Nigel, Gilles). Get frequent feedback from clients, supervisor and professionals.

Title	<i>Data Storage</i>
Severity	medium
Description	Loss of data due to hardware failures or hardware theft.
Mitigation	Storing all project related information on a network machine with daily backup routines (see subsection 3.3.1).

Title	<i>Testing difficulty</i>
Severity	medium
Description	Testing environment, guidelines and timeframe are unclear.
Mitigation	Test early and involve the client into the testing.

Title	<i>Commitment</i>
Severity	low
Description	Lack of client commitment.
Mitigation	Involve client in project work and frequently show results to highlight project benefits.

Title	<i>Integration time</i>
Severity	low
Description	Estimated integration time is longer than expected.
Mitigation	Use of shared version control system for continuous integration of all developers' work (see subsubsection 3.3.1).

Title	<i>Communication</i>
Severity	low
Description	Communication problems in development team, which is dispersed among several sites.
Mitigation	Use of a source code version control system to enable continuous collaboration. Make use of a dedicated project space to achieve developer collocation.

Title	<i>Requirements</i>
Severity	low
Description	Stated requirements might not match the customers' desires for the system.
Mitigation	Evaluating requirements with customer in consequent meetings. Ensure understanding of requirements by prototypical implementations and customer feedback.

5.2 Feasibility

General View Due to the high complexity of this project and the short timeframe, the project members hope to be able to keep the risks low and to overcome the possible hindrances. But since this is an interesting project and the project members have a high motivation, they are confident to achieve the stated goal.

Economical view Project task is to solve a real world problem coming out of a development effort at the James Cook University in Townsville Australia. The client is the computer science branch at the JCU that is currently working on the sensor network. (better name project description) Since the project is part of a student assignment inside a project class, high commitment of project participants can be assumed. Additionally the costs for this project do not involve any costs to pay the project members and are therefore limited to hardware costs of development equipment and costs of communication with the client. Therefore, the JCU will lessen the costs to develop the needed functionality by freeing internal resources by outsourcing the described tasks of section 2 & 3 of this document.

Strengths	Weaknesses
<p>Expertise The project members are familiar with all needed aspects of the development environment. This includes Linux, Eclipse, Subversion and Java. Therefore no introductory training for these topics is needed at the start of the project.</p> <p>Motivation The project members are highly motivated.</p>	<p>Project insight Project members had no prior contact with the project environment. This might cause problems in understanding the clients needs.</p>
Opportunities	Threats
<p>Deadline The project is part of a bigger project that has a due date which corresponds with the deadline of the subproject. This might provide additional client commitment.</p> <p>Client location The clients are located on campus, which makes it easy to come along.</p>	<p>Deadline Since two deadlines fall together, the client might be too occupied to participate in testing and frequent meetings.</p>

Table 1: SWOT matrix

References

- [BKS] BRAUN, Jochen ; KNIRSCH, Andreas ; SEEMANN, Andreas: *SAL-T Code Home*, <http://code.google.com/p/sal-t/>, Last Checked: 01.09.2008
- [Cen] CENTER, San Diego S.: *Storage Resource Broker Home*, <http://www.sdsc.edu/srb/index.php>, Last Checked: 02.09.2008
- [Fou] FOUNDATION", Eclipse: *eclipse home*, <http://www.eclipse.org/>, Last Checked: 01.09.2008
- [Gig] GIGAN, Gilles: *SAL project overview*, <http://plone.jcu.edu.au/dimsim/Members/gillesgigan/sal/sal-overview>, Last Checked: 02.09.2008
- [HPR] HPRC, JCU: *Hermes*, <http://www.hpc.jcu.edu.au/projects/data-activities/wiki/Hermes>, Last Checked: 02.09.2008
- [Ini] INITIATIVE, The Open Source D.: *Data Turbine Home*, <http://http://www.dataturbine.org/>, Last Checked: 02.09.2008
- [PP03] POPPENDIECK, Mary ; POPPENDIECK, Tom: *Lean Software Development: An Agile Toolkit*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2003. – ISBN 0321150783
- [Wys06] WYSOCKI, Robert K.: *Effective Project Management: Traditional, Adaptive, Extreme*. 4th. Hoboken, New Jersey : Wiley, 2006. – ISBN 978-0-470-04261-8