Individual Project Plan
OSGi in Distributed Robotic Systems

Ben Coughlan
Australian National University
Canberra, Australia

ben.coughlan@anu.edu.au

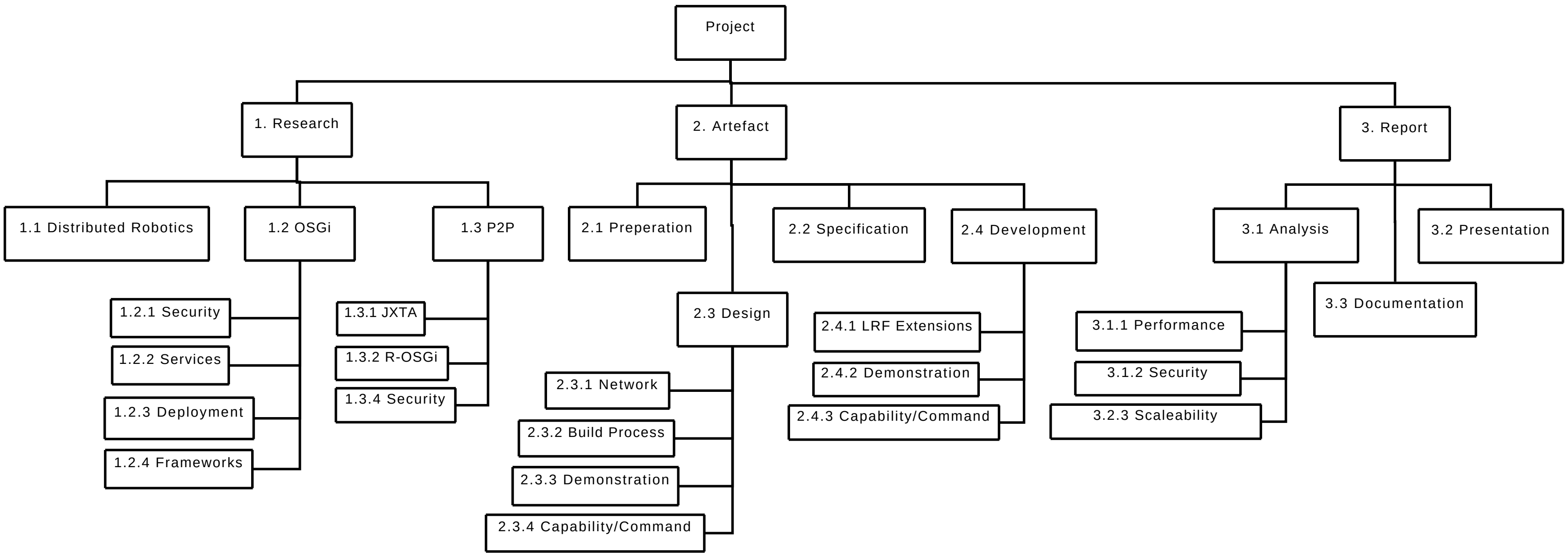
March 26, 2009

1 Project Description

This project aims to demonstrate the capabilities of an OSGi framework at providing a dynamic module support framework as well as transparent distribution of service components. Combining this technology with a peer to peer approach to network topology such as JXTA may be shown as a powerful approach to managing and controlling complex systems of agents.

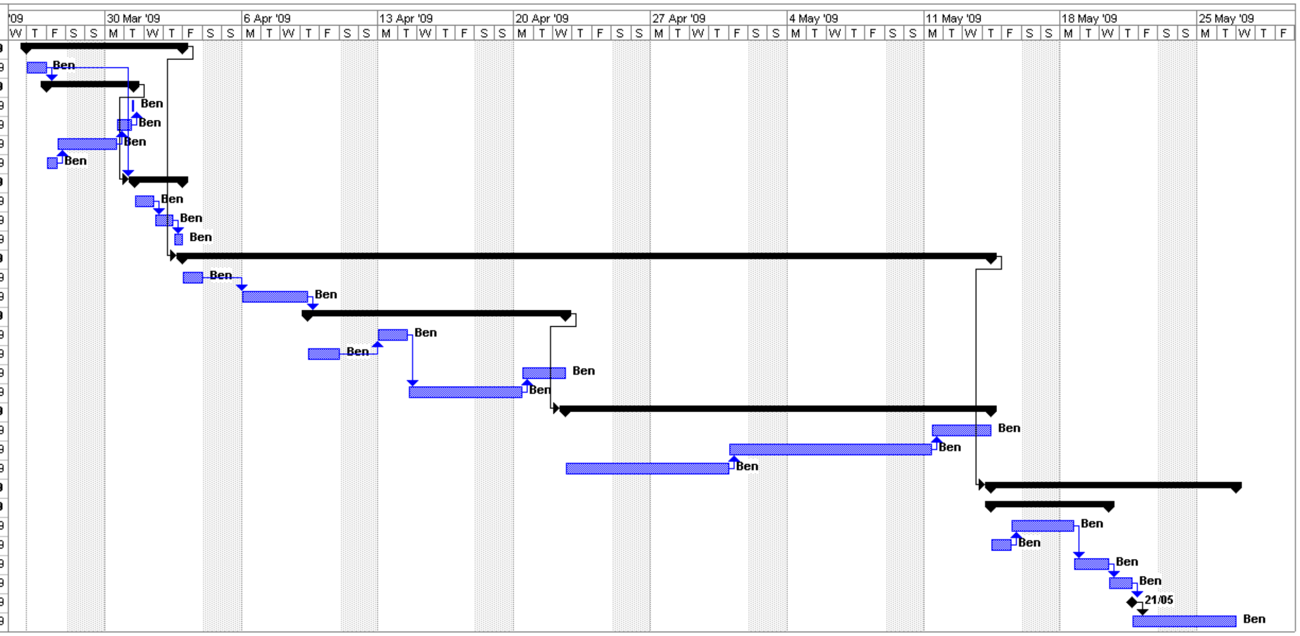
The project will be divided into three distinct phases. Starting with a research phase, all the technologies identified as being useful will be investigated. Then there will be a development phase which will follow a standard software development life cycle. Finally, there is an analysis and reporting phase where I will draw all my conclusions.

2 Work Breakdown



3 Schedule

ID	Task Name	Work	Start	Finish
1	**Research**	**136 hrs**	**Thu 26/03/09**	**Thu 2/04/09**
2	Distributed Robotics	8 hrs	Thu 26/03/09	Thu 26/03/09
3	**OSGi**	**40 hrs**	**Fri 27/03/09**	**Tue 31/03/09**
4	Security	3 hrs	Tue 31/03/09	Tue 31/03/09
5	Services	4 hrs	Mon 30/03/09	Tue 31/03/09
6	Deployment	8 hrs	Fri 27/03/09	Mon 30/03/09
7	Frameworks	5 hrs	Fri 27/03/09	Fri 27/03/09
8	**P2P**	**40 hrs**	**Tue 31/03/09**	**Thu 2/04/09**
9	JXTA	8 hrs	Tue 31/03/09	Wed 1/04/09
10	R-OSGi	8 hrs	Wed 1/04/09	Thu 2/04/09
11	Security	4 hrs	Thu 2/04/09	Thu 2/04/09
12	**Artefact**	**672 hrs**	**Fri 3/04/09**	**Thu 14/05/09**
13	Preperation	8 hrs	Fri 3/04/09	Fri 3/04/09
14	Specification	25 hrs	Mon 6/04/09	Thu 9/04/09
15	**Design**	**154 hrs**	**Thu 9/04/09**	**Wed 22/04/09**
16	Network	12 hrs	Mon 13/04/09	Tue 14/04/09
17	Build Process	15 hrs	Thu 9/04/09	Fri 10/04/09
18	Demonstration	20 hrs	Mon 20/04/09	Wed 22/04/09
19	Capabiliy/Command	30 hrs	Tue 14/04/09	Mon 20/04/09
20	**Development**	**250 hrs**	**Wed 22/04/09**	**Thu 14/05/09**
21	LRF Extensions	25 hrs	Mon 11/05/09	Thu 14/05/09
22	Demonstration	50 hrs	Fri 1/05/09	Mon 11/05/09
23	Capability/Command	50 hrs	Wed 22/04/09	Thu 30/04/09
24	**Report**	**170 hrs**	**Thu 14/05/09**	**Tue 26/05/09**
25	**Analysis**	**64 hrs**	**Thu 14/05/09**	**Wed 20/05/09**
26	Performance	12 hrs	Fri 15/05/09	Mon 18/05/09
27	Security	8 hrs	Thu 14/05/09	Fri 15/05/09
28	Scaleability	12 hrs	Mon 18/05/09	Wed 20/05/09
29	Presentation	12 hrs	Wed 20/05/09	Thu 21/05/09
30	Final Presentation	0 hrs	Thu 21/05/09	Thu 21/05/09
31	Documentation	25 hrs	Thu 21/05/09	Tue 26/05/09



4 Work Descriptions

4.1 Research

These work packages make up the research phase of the project. This will involve investigating the technologies that have been identified as well as the key concepts required for distributed robotic systems.

4.1.1 Distributed Robotics

This is to give background in the field and a context for study into relevant technologies. This will be conducted by discussions with people involved in the area and potential users.

4.1.2 OSGi

Research into discovering the capabilities and features of the OSGi standard and its implementations.

4.1.2.1 Security Research the various attack vectors in an OSGi framework. In a dynamic module framework, the biggest risk is having one module "crash" and damage the safe operation of other modules in the framework. This also leads to the potential for malicious code running as either a legitimate module, or as the result of an error in another module.

Research will also be done into the security mechanisms within an OSGi framework, as well as discussion about security mechanisms found in a standard Java virtual machine.

4.1.2.2 Services The most powerful feature of an OSGi framework is allowing bundles to provide services to other bundles. While this is fairly simple concept, it warrants detailed research to understand the uses and limitations of this feature.

4.1.2.3 Deployment OSGi frameworks may enforce a design paradigm that should be familiar to all software developers, but the practical implementation of OSGi bundles requires a different approach to construction and deployment. While this will not include making user friendly installers etc., it will cover things like bundle repositories, on-line deployment and appropriate build utilities.

4.1.2.4 Frameworks I am currently aware of four implementations of the OSGi standard. Namely; Equinox, Felix, Knopplerfish and Concierge. These implementations vary in size and features as well as intended use. Some work will go to identifying the differences between them and what situations (in this project's context) they would be suited for.

4.1.3 P2P Networks

Given the dynamic nature of OSGi is the inspiration for this project, basic network infrastructure is a major bottle-neck in the ability to demonstrate the desired outcomes. There will be some investigation into alternative network arrangements to compliment OSGi's dynamic nature. The only truly scalable alternative is some form of peer to peer network.

4.1.3.1 R-OSGi Remote OSGi is a bundle installed in an OSGi framework that copies services across some communication medium into another instance of a framework. This allows consumers of those services to access a remote instance completely transparently. This provides a very powerful mechanism for distributing a dynamic system.

The R-OSGi project is still quite young and research will be needed to identify its capabilities and limitations.

4.1.3.2 JXTA
JXTA is a collection of P2P protocols aimed at completely managing and utilising a network of peers. It can likely be used as a transport layer for a higher level, OSGi based, communication mechanism. There will also be research into JXTA's suitability for use over a serial connection, as well as alternative network stacks to TCP.

4.1.3.3 Security
This will look into the security concerns of JXTA and R-OSGi. This will mainly identify the conceptual strengths and weaknesses of these protocols and investigate what mechanisms they have in place for providing secure, encrypted communications.

4.2 Artefact

This work will relate to the demonstration software and detailed designs produced by this project.

4.2.1 Preparation

Having identified reasonable ways to build and deploy software items in 'Deployment', this will focus on building the infrastructure required to produce those software items. This will require installing relevant software, providing repository locations and configuring a (hopefully) standard build process.

4.2.2 Specification

The specification of the deliverable artefacts will be in two parts. Firstly; a specification of the intended system and appropriate concepts for development within that system. Second; the specification of a demonstration which uses that system.

4.2.3 Design

The design will follow the same idea as the specification, and will produce a detailed design document for interested parties.

4.2.3.1 Build Process
A solid build process is required for efficient development. Given the differences in deploying an OSGi based application, the build process should be decided early.

4.2.3.2 Demonstration
This will require the design of all components specific to demonstrating a proof of concept and will essentially be a throw-away document with little value beyond the development of a proof of concept.

4.2.3.3 Network
This is a small task to design the various approaches to a network infrastructure. Basically how to combine various P2P technologies with TCP/Serial/any other medium of interest.

4.2.3.4 Capability & Command
This part of the design will require the most thought, and will probably make up the bulk of the design artefacts. Here is where a typical bundle will be defined as well as how it is expected to interact with other bundles in the system. There should be thought towards AI concerns to allow planning as well as a 'chain of command' that will determine which agents can control/access other agents.

This design document should be created with a long term project in mind, but with only a small subset to be implemented this semester.

4.2.4 Development

Here are all the development tasks for implementing the above design. These will focus on a proof of concept implementation and will only be a subset the the specification. Core components should be produced as if the project were being continued past the end of this semester.

4.2.4.1 LRF Extensions/Modifications The Linux Robotics Framework project forms a similar abstraction layer at a lower level. This includes a set of drivers for single hardware components as well as a way of combining them to access complex components from a simple API. This work will develop a way of accessing LRF libraries from within an OSGi bundle.

4.2.4.2 Demonstration This will be a collection of virtual agents to use for a demonstration. These agents will perform very simple tasks within the demonstration world and are not expected to be re-used after this projects completion.

There will also be a requirement to tie together various other technologies such as OSGi and JXTA for the demonstration. It is likely this work will be specific to the demonstration.

4.2.4.3 Capability/Command This will be the main part of the software artefact that is developed from scratch. This should be implemented and tested to a reasonable standard as it will form the basis of any continued work on this project.

4.3 Report

This work is to document and present the system developed.

4.3.1 Analysis

Here is the evaluation of the system with a focus on its practical implementation.

4.3.1.1 Performance Analyze how certain tasks in the system perform. Tasks within a single OSGi instance will really just be testing the performance of the underlying Java virtual machine. For this reason, performance analysis will focus on tasks that are distributed over many agents.

4.3.1.2 Security Consider the aggregate security concerns of the involved technologies, identified in the research phase. Also consider any changes to these concerns given the specific implementation.

4.3.1.3 Scaleability Consider how the underlying P2P and OSGi technologies scale as more agents are involved in the system. This will focus mainly on the number of communication channels and the impact on agents assigning tasks.

4.3.2 Presentation

Create a presentation explaining the outcomes of the project and organise a demonstration of the software system developed.

4.3.3 Documentation

Compile all my findings into a final report.

5 Initial Risks

5.1 Technologies

5.1.1 R-OSGi cannot perform as expected

Affects: Design [2.3.0] & Development [2.4.0]
Probability: Moderate
Impact: High
Mitigation: The R-OSGi bundle is the main mechanism for distribution of tasks across the system. If this bundle cannot function as expected, the projects focus will shift on adapting this bundle or creating a new bundle to provide this function. R-OSGi will be verified very early in the project so this change can be incorporated quickly if required.

5.1.2 JXTA cannot perform as expected

Affects: Design [2.3.0] & Development [2.4.0]
Probability: Low
Impact: Low
Mitigation: JXTA is a popular project and has been adapted to many situations, so this risk is fairly low. Its use in this project is really a convenience so if it cannot provide the service required, then a simple workaround can still make the demonstration work.

5.2 Equipment

5.2.1 Project server crash

Affects: Development [2.4.0]
Probability: Low
Impact: High
Mitigation: The project server will be required for smooth, reproducible builds. If the server was lost, it would have a significant impact on the development phase. Server backups will be performed stored offsite, and other servers will be available if the main one was unavailable.

5.2.2 Some embedded equipment may not be available for development or demonstration

Affects: Development [2.4.0] & Presentation [3.2.0]
Probability: Moderate
Impact: Low
Mitigation: The development and the presentation do not rely on any equipment being available. If equipment is not available, the demonstration will just use virtual agents. This must be decided in the design phase.

5.3 Other

5.3.1 External commitments prevent time spent on this project

Affects: Report [3.0.0]
Probability: Moderate
Impact: High
Mitigation: Stock up on caffeinated beverages and other stimulants. Once lost, time can only be recovered by sleep depravation. Some parts of the projects scope may trimmed if there is not enough time.

5.3.2 Required third-party project loses support

Affects:	Report [3.0.0]
Probability:	Low
Impact:	Low

Mitigation: Within the duration of this semester, a third party project being scrapped will have little impact. However, copies of libraries and some documentation should be copied locally incase a website becomes unavailable.

5.3.3 Uwe subtracts marks for a lack of 'real-time' considerations

Affects:	Report [3.0.0]
Probability:	Moderate
Impact:	Moderate

Mitigation: Uwe's concern is if this system is practical outside of my proof of concept demonstration. While I don't expect my incarnation of it will be, the concepts certainly can. Be sure to write at least one paragraph in the final report describing real-time considerations in the context of this project in practical, real-world environment. This should include some suggestions on how to modify the project artefacts to suit real-world applications.