Distributed Integration Testing

OSGi Testing Framework for testing lifecycles

Phuoc Tran

De Steeg

5-6-2012

luminis

Conversing worlds

Graduation Paper

Distributed Integration Testing

Author
Phuoc Tran
2125609
HBO ICT & Software Engineering

Company
Luminis Technologies
Hoofdstraat 130
6994 AK De Steeg

Mentors
Paul Bakker
Luminis Technologies

Marcel Offermans
Luminis Technologies

School Mentor
Andries Kasper
Software Department of Fontys Hogescholen ICT in Eindhoven

Date :

Signed By: Paul Bakker and Marcel Offermans

DOCUMENT REVISION

Date	Author	What
4-06-2012	Phuoc Tran	Exported to Word v.9
5-06-2012	Phuoc Tran	Changes after feedback of mentors v.46

PREFACE

This document is written for the Distributed Integration testing project. The intern is working on the graduation for the program Software Engineering that is held at The Fontys Hogescholen in Eindhoven. The project is done at Luminis Technologies in Arnhem in the period of February 2012 till the end of July 2012. This document describes the process of the work that I've done at Luminis Technologies.

I want to thank Marcel Offermans and Paul Bakker for giving me the opportunity to do the graduation project at Luminis Technologies and I also want to thank them for all their support. I also want to thank Jan Willem Jansen for his contributions to the project. A thank you goes to my tutor Andries Kasper for his guidance. Also I want to thank the people of Luminis for their kindness and co-operation.

Phuoc Tran

De Steeg, June 2012

INHOUD

SUMMARY

Distributed Integration testing is a graduation project that has been done at Luminis Technologies. Luminis Technologies a division of the Luminis Group. The members of Luminis Technologies are developing innovative software components. The Luminis Group is a platform/network organization and is divided in six divisions. Each of the divisions has its own expertise and the group expects that each division maintains this expertise by increasing its knowledge. Luminis Technologies develops modular software components that can be assembled into dynamic applications that are distributed in public and private clouds, spanning a number of nodes. These components are built using a Java based framework called OSGi and distributed onto the cloud nodes using a software distribution application called Apache ACE. Both are part of a larger open source initiative called Amdatu, a platform for building cloud based, dynamic applications. We need to test if the distribution of the OSGi components onto multiple nodes and the actual components themselves are working correctly.

The project consist out of three members. The members have their own responsibility and role in the project method SCRUM. The reason for using SCRUM is that we can step by step solve a part of the solution. This is been done in an iterative way. This means that after each iteration we are going to refine the requirements and design of the solution. Another key point of SCRUM is the standup. A standup is a daily meeting that allows the project team to get up to date with the progress of the project. The team will tell what their progress was and what they are planning to do. In the project initiation document the intern explains that the project will be managed with a combination of SCRUM and PRINCE2, however when the project progressed it became less PRINCE2 and more SCRUM. The school of the intern set a requirement to use the Tien Stappen Plan method for managing the graduation. The method divides the graduation in 10 phases. Each phase works as a checklist. After a phase you are going to check if you got the deliverables. You go to the next phase if all the deliverables are checked.

The research is been done to find a possible way to test the integration of OSGi application that are distributed on multiple nodes by researching and comparing existing solutions. To do this you need to know what the current situation and problem. After the current situation and the problem were define the research question was divided into several domains. Each domain got its own questions that are answered in the research. These domains were: requirements, integration tests, OSGi, distribution of tests and current test framework. The research resulted in a possible solution to test the integration of OSGi applications. Apache ACE is going to be used to distribute the tests that Arquillian is deploying to execute on a node or, as ACE calls it, a target.

The design of the test framework changed when the parts of the test framework were discussed in the project team. The reason for this was that the vision of the users how the test framework should look like evolved. This resulted in a design that got more refined after each iteration. After a part of the test framework was designed, the next step was to implement the part of the test framework. This is all done with Java tooling. The implementation was discussed with the users and they gave feedback how to improve it or told the intern to start the next part of the test framework. The result of the project is a test framework that allows a developer to test the integration of OSGi projects on multiple nodes.

INTRODUCTION

Luminis Technologies is working on projects that can be distributed over a dynamic amount of nodes. A node is an environment that the code of the project is executed. Luminis Technologies wants a way to test these projects. The projects are OSGi applications/services and can be distributed to any amount of targets. Testing the traditional way doesn't cover the distribution of these projects and you cannot be sure if the OSGi projects will be able to run once they are deployed.

This aim of the project is to find a possible way to test the distribution and the code that is running on multiple nodes. Research has been done to find the possible ways to be able to test the distribution and the execution on the nodes. The result of the research is a design for a possible solution that can test the distribution and execution. After the research has been done, the solution needs to be build to support the conclusion of the research.

The design is explained in the architectural overview document. For the information of how to the project is started please check the Project Initiation Document. The Project initiation document describes how to project is been approach, what the roles were of the project members, what project methods are used and what the initial planning was. For all the information of the research I will refer you to the research document. The research document covers the approach of the research, what the results are from the analysis of the problem and possible solutions. The requirements are explained in the requirements document.

This document is also a 'summary' of these documents and is divided in several chapters:

- Chapter 2 explains the information about the company Luminis were the project took place.
- Chapter 3 explains what the current situation is and what the assignment is.
- Chapter 4 explains how the project is approach and what project methods were used.
- Chapter 5 gives a summary about the research that has been for Luminis Technologies.
- Chapter 6 gives an overview of the process of the project.
- Chapter 7 contains the conclusion of the project.
- Chapter 8 is the retrospective of the project.

Attachments that are included:

I. Project Initiation Document
II. Research Document
III. Requirements Document
IV. Architectural Overview

GLOSSARY

Term	Explanation
Bundle(s)	A bundle is jar file with specific metadata that makes an OSGi framework recognize it as a valid software component
Clusters	A cluster is group of nodes that are working together
I/O	Input/Output.
Jar	The code of the project is packaged in an archive called a jar
JUnit	JUnit is a framework for executing unit tests in Java
Maven	Apache Maven is a tool for building Java code and managing its build-time dependencies
Node(s)	A node is where the applications/services are running.
OSGi	OSGi is a framework for developing and running modular and dynamic applications in Java
Target	Another term for a node, mostly used by Apache ACE
Test cycle	The test cycle is a process that is started by JUnit

2. THE COMPANY LUMINIS

This chapter will explain what type of company Luminis is and how it all began, the vision of Luminis and where the graduation project takes place.

Luminis is a company that develops services and software for its customers. The software and services can be mobile application or entire cloud solutions. Luminis is striving to be innovative. To do this Luminis is collaborating with customers to find solutions that are innovative. Luminis is making software and servies that 'convert' the current situation in a innovating way. A great example of such a project is "Leren-Op-Maat". The purpose of the project is to give a students a personalized way of learning. The students use an e-learning environment. This environment contains multiple ways to learn a subject and based on the choices of a student the system can decide which way is the most suitable for this student. It is not just the idea of that personalized learning that makes it innovative. The technical aspects are also part of the innovation. "Leren-op-maat" project is build on a platform that allows projects to be build into full blown cloud applications that support advanced use cases as provisioning and scaling without introducing a development model that is complex

Luminis is a networked company or rather a platform that is divided in several cores. The platform is divided in several "cores" each with their own expertise. The cores are basically small company that act on their own. These cores are :

- Luminis Apeldoorn
- Luminis Arnhem
- Luminis Closure
- Luminis Technologies
- Luminis Rotterdam
- Thesio

Each Luminis core has its own expertise and the platform is expecting that each core maintains the expertise by increasing its knowledge about the expertise and share this knowledge with the platform. This results in a constant self-improving platform. Luminis was founded in 2002 with as vision: *"The art of Software Engineering". They have a vision that software is craftsmanship. "We believe that software development is a craft; it has many of the characteristics represented by a medieval guild – as Richard Senett described software developments as a craft in his book, "The Craftsman", and as Freeman Dyson wrote about "Science as a craft" in his essay."*
"Despite the emergence of Microsoft and other large producers, software development remains largely a craft. The enormous variety of specialized applications means that there will always be room for individuals to create new software using their unique combination of knowledge and expertise. Niche markets prove time and time again that small companies have a role to play. The craft of writing software will not age, but continues to expand itself." http://www.luminis.eu/en/expertise/software-craftsmanship/

2.1 ORGANIZATION

The organization structure is visualize in the following diagram:

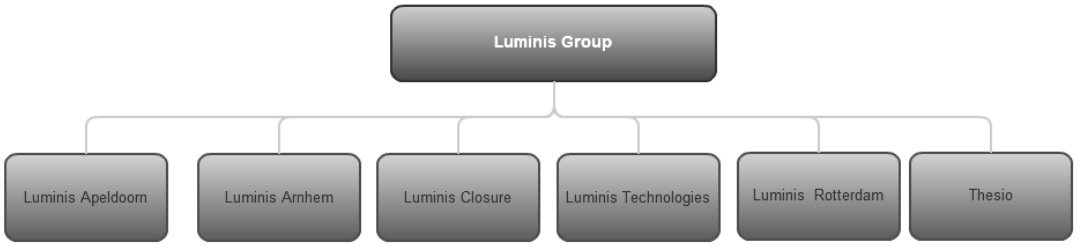

Figure 1 Organization

The project is done in the core called Luminis Technologies. The website states that Luminis Technologies : "*is the software technology and support division of the Luminis group; we develop, distribute and support a suite of innovative software components that enable organizations and developers to exploit the full potential of an interconnected, interoperable world*".

Technologies that the division are working on are provisioning in OGSi. Provisioning is term that is used for a process to distributed application and resources a network of devices/nodes or targets. Luminis Technologies is working on a OSGi provisioning system that has been donated to the Apache Software Foundation. The Apache Software Foundation provides legal, organizational and financial support for open source projects that are done in the community. The system is called Apache ACE and allows you to provision your OSGi applications and services to nodes that are managed by Apache ACE.

3. THE PROJECT ASSIGNMENT

This chapter will explain what the initial situation is and what the actual problem is. After that the assignment will be explained, what the project constrains are and finally what the desired outcome of this is.

3.1 INITIAL SITUATION

Luminis Technologies uses OSGi and Apache ACE for its development and this allows them to build services/applications. This results in application that are distributed over a numbers of nodes. The life cycle of these applications is dynamic. When the applications are deployed on the node, the node is going to look if the applications can start. If the application needs other services that aren't running, the application will not start. If the services are up and running then the nodes will start the application. This means that there should be a way to test the distribution of these applications and it needs to test if the applications are working. The solution should allow us to write tests that test the life cycle of the application/services. Another thing that needs to be tested is the life cycle of the nodes. When the nodes are working together as a cluster and one node is shutdown, what will happen with the services and applications on this cluster?

3.2 ASSIGNMENT

The assignment is to see if there is a possible way to test the integration of these services on multiple nodes. This means that a research needs to be made to see what exactly the problem is and to find a possible solution for this problem. In order to do this some technologies are going to be researched.

The first part of the researching technologies is finding a possible way to distribute the tests. This is done by researching the systems that can distribute applications.

The second part of researching the technologies is find a way to execute the test. This is done by looking at test frameworks that are capable of executing on multiple containers. There is already a set of requirements at the beginning of the project that the users defined however these requirements need to be analyzed and interviews are going to be held in order to be sure that all the requirements are gathered. The next thing that has to be done is to find a solution that can be used by the users.

The assignment contains the following points that are going to be done in the project:

- Analysis of the current situation, a definition of the problems and a strategy for finding its solutions.
- Gather the additional requirements by interviewing the users.
- Prioritize the requirements of the beginning of the project and the additional requirement of the interviews
- A prototype of the test framework needs to be built to show if the research conclusions are correct.

3.3 OUTCOME

The desired solution is a test framework that allows us to distribute the test on multiple nodes and execute them. It should be able to test the life cycle of bundles and targets in an environment that is similar to the production environment. It has to be user friendly meaning that it needs to be easy to setup. The test framework should be able to test multiple targets.

4. THE PROJECT APPROACH

This chapter explains how the project is approached, what the project team looks like and what project methods were used

How these things should be done is explained the Project Initiation Document (PID). This document explains what the planning is, what project methods have been used and what phases the project has. To explain the document, the next paragraph will give a quick overview what the project looks like.

The project team has got three members. Two of those members are the users and the other member is the intern. This is visualized below. The roles and the responsibilities are explained in the PID. As you can see there is also a external member which is the tutor from the school. He is responsible for evaluating the documents.

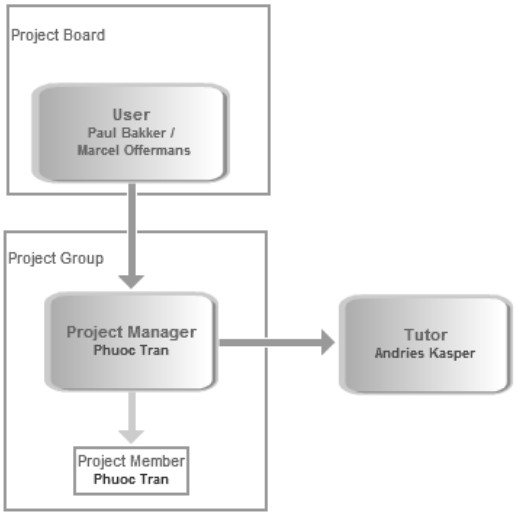

Figure 2 project team

The project is managed by SCRUM. SCRUM is an agile project method. This means that the requirements and the solution will evolve with the progress of the project. A SCRUM team contains a couple of roles:

- SCRUM master, which is managing the scrum process
- Product owner, which represent the customer
- Project Manager, which are the persons that are responsible for the product.

In the following table you can see what role each person got.

Name	Role	Interest	Responsibility	Contact details
Marcel Offermans	User	Products/results of a sprint	Giving feedback on the results and products	marcel.offermans@luminis.eu
Paul Bakker	User /SCRUM Master /Product Owner	Products/results of a sprint	Giving feedback on the results and products	paul.bakker@luminis.eu
Phuoc Tran	Project Manager	Managing the project	Maintaining contact with the users	phuoc.tran@luminis.eu
Phuoc Tran	Project Member	Requirements	Realization	phuoc.tran@luminis.eu
Andries Kasper	Mentor	Status of the graduation	Giving feedback on the PID and the graduation paper	a.kasper@fontys.nl

SCRUM framework is used in this project for its iterative based way of working. The project uses technologies that are unknown for the intern. It is hard to dive into it and make finalized design and make one implementation, because the intern is not yet familiar with the technologies. When using SCRUM we can take a small bit of this unknown and spend a fixed amount of time and discuss the results with the project team. You can see it as a puzzle is solved piece by piece. Communication is a key factor for the project and having a daily standup made it possible to inform the progress to every project member so that we are on the same page.

In the beginning everything can be unclear and the requirements are global and when the project progress, things get much clearer and which could lead to changes in the requirements. Using a less iterative way of working doesn't allow to revisited requirements. This is possible risk because requirements will still be global. We worked in an iterative way to make things clearer. This means when a piece of the solution is solved, we are going to discuss this. In this discussion the intern explains how it works and the users give their feedback how to improve it. This way the solution will get better after each discussion. This discussion can be held online or offline. Because most of the members are working from home, the quickest way to get everybody up to date is to setup a meeting with the usage of the internet. The online meeting, which we call a standup, is held every day. In the standup the intern tells what to progress is where he is working on and what he is planning to do that day. The great thing about those meetings is that the users are involved. This means you have contact with your users every day. You can discuss how they want each part of solution to work.

Another positive thing about working this way is that you and the users are on the same page. No longer do you get the classic programming problem that you when you released the final product the user isn't satisfied. When an important part of the solution is solved and implemented an offline meeting is held. The intern presents it to the users and the users give feedback. The reason why we work this way is that at the beginning of the project both technologies and the exact requirements were unknown for the intern and it is unrealistic to try and plan everything upfront in this situation. You need to manage it. Step by step the intern went deeper into the technologies and got more feedback from the users about how to improve the solution.

A risk of working this way is that the planning is not clear in the beginning which could cause the project to get delayed. You don't know how much time it will take to make a part of a solution. To prevent this we have our daily standup which allow us to check the planning and adjust so that we are in control of the project.

4.1 PLANNING

The initial planning was:

Weeknr	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
PID																
Research plan																
Requirements Document																
Research answers																
Functional Design																
Technical Design																
Research Results																
Prototype																
Test Document																
Documentation																
Graduation paper																

Back then the PID stated that this project should use SCRUM with PRINCE2. However due to the preferences of the company, it became more SCRUM and less PRINCE2. The way PRINCE2 manages a project is to clearly define what needs to be done when there are changes. It is very strict and will cause a lot of overhead when there are a lot small changes in the project. The requirements are based on how the user thinks that the test framework should work. The user got a vision of how the test framework should work but this can change when the intern presents a piece of the solution. In PRINCE2 this means that the functionality can change and that a change management process needs be started. With SCRUM you are expecting that there are going to be changes which means that using SCRUM you don’t spent on change management. But you are more focused on what are you be able to do in the amount of time.

This project couldn't be done in a less iterative way because in the beginning we did not know what the results of the project would be. Requirements can be vague and when the project progresses, the requirements will be refined by revisiting the phases design and implementation. In a non-iterative method you aren't allowed to revisit the phases, which may result into a solution that is not working according the final requirements because the initial requirements were vague or even impossible to achieve. In the project plan you can see that there is this huge blob. The design and implementation phases are done simultaneously. In the planning you can't see when a phase of Prince2 ends and a new phase is started. The project is not using the PRINCE2 phases. What still stands are the deliverables. These are in stated in the planning and are: the PID, requirements, research and the architectural overview which contains the designs.

4.2 TIEN STAPPEN PLAN

My school has set a requirement that the graduation project is using the "Tien stappen Plan" (TSP) [tsp] method. The method divides the graduation in 10 phases.

- External Orientation
  - In this phase the intern should be looking for information about the company. Specific what are the key points of the company?
- Intake interviews
  - In these interviews the intern should get more information about what the users want and what the users are expecting of the intern
- The kick off
  - The intern should be able to describe the situation and should be able to define the assignment.
- Analysis
  - The PID will be written in this phase
- Feedback and contract
  - The PID will be discussed with the users and the go/no go is decided
- Planning and project organization
  - In this phase the organization needs to described and the planning should be made
- Development
  - In this phase the development of the solution or the research should start
- Results
  - In this phase the intern should have released the solution and the intern should have the research conclusion and advice.
- Implementation at the users
  - In this phase the solution should be implemented in the environment of the user
- Closing the project
  - Preparing the for the graduation presentation.

When working this way you are basically working with a checklist. After a phase you check if you have got the deliverable and you will go to the next phase. In this project we don't know how the planning will go because we don't know how the research will progress therefore we couldn't go and finalize the planning in phase 6. What is important to know that phases 7-9 will take the most of the time. So how did I implement (TSP) in this project? As you can see the first 5 phases are orientation. The results of these phase are written in the PID. I took less details to describe the company, because the solution is tooling that allows the users to increase the quality of the software that the company produces. Phase 7-8 describes the development and releases. In these phases we worked with SCRUM for the development and research. The phases are done in the iterative way. The phases are redone to refine the requirements and the design. In phase 9 the solution should be implemented in the environment of the user. This means for the project that the users are going to use the solution and give the feedback. Finally phase 10 is the preparation for the graduation presentation.

5. THE RESEARCH OF THE PROJECT

This chapter is about the research. It explains how the research is approach and what the results and conclusions are.

The research is the main part of the project. The assignment was to analyze the possibility of integration testing of OSGi applications that are distributed over multiple nodes by researching and comparing existing solutions. In order to have a research topic we need to have a main question concerning the problems in the current situation.

5.1 THE APPROACH

The research was approach by using a research technique from "Verschuren en Doorewaard". The key of this technique is the research model. This model divides the research question into domains and the model also divides the research in phases. This model is included in the research plan which is just an initial plan to approach the research.

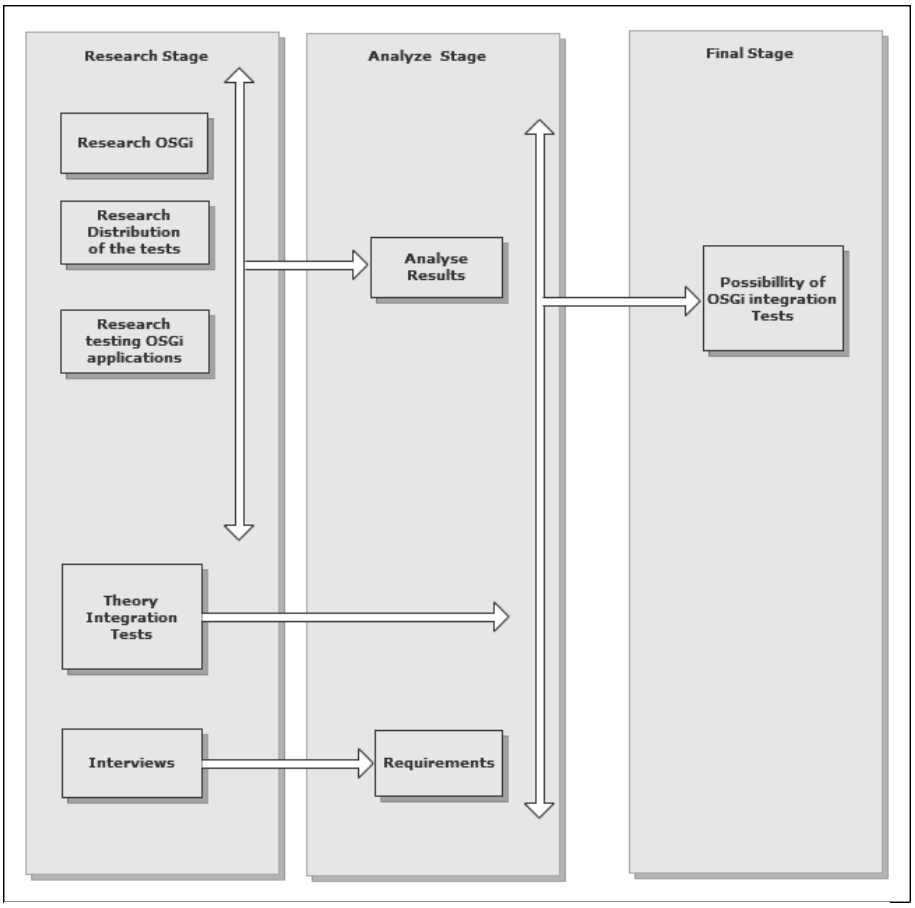

Figure 3 Research model

Each of these domains can be researched by answering question. To answer the questions about a technology domain, knowledge of the technology need to gathered by using it.

5.2 DOMAINS

USER REQUIREMENTS

At the beginning of the project there was a list of requirements that the users have defined. These requirements needed to be prioritized so that the most important ones are implemented first. The requirements were pretty clear however it's always a good thing to check requirements by interviewing the users. This allows me to get some extra information and I can tell them what my vision of the problem is. The people that were interviewed are users that are developing with OSGi in an environment that allows applications to be dynamically distributed. These people are chosen because they are working daily with the technologies in other projects in Luminis Technologies and they all experience the problems with testing in a multiple node environment. The results of these interviews were then converted in set of requirements. These requirements are prioritized and each of these requirements contains one or more user stories. The questions were:

- What is the problem with the current situation of testing the OSGi bundles that are distributed over multiple nodes?
- How do you see the solution?
- What makes a test framework user friendly?
    - How do you want to deploy additional bundles?
    - How do you want to write a test?
- How do you want to use the test framework?
    - How do you want the result back?

This resulted with the set of requirements in the beginning of the project in a list of requirements prioritized in a MoSCoW list

Must

- You should be able to write the test like a unit test, using familiar tools such as JUnit.
- You should be able to test on a existing target.
- You shouldn't spend more time on configuring the test framework then you would with plain unit tests.
- The test framework must be just one single jar file that a developer has to include in a project.
- Test classes and bundles that are going to be tested are deployed in separate bundles.
- A developer should be able to specify a bundle that has to be running on a target by a combination of its symbolic name and version.
- A developer should be able to specify the target of deployment by a single line of code.
- A developer must be able to specify the location of the Apache ACE server by configuration.
- A developer should be able to run a test by using Maven, Ant or BndTools.
- A developer should be able to include resources like configuration files.

Should

- A developer should be able to start and stop a target with a single line of code.
- A developer should be able to test on multiple targets.
- A developer should retrieve information about the cause of failing tests.
- Should be able to provide configuration both from configuration files and code.

Could

- A developer should be able to use Apache ACE embedded.
- A developer should be able to test the life cycle of a target by manually creating a target, identifying it in Apache ACE, deploy a distribution and undeploy it.
- A developer should be able to debug a target with debugger.

Nice to have

- Look up stored testcases by looking at a set of bundles
  - Stored testcases are testcases that are predefined. These testcases can be executed based on the set of bundles that are deployed.

INTEGRATION TESTING

The main purpose of integration testing is that you test the parts of the software combined to see if these parts of the software are working with each other. The can be done in two ways:

- Top down: You start by testing on the highest level of an application and work your way down.
- Bottom up: You start by testing the core of the application and work your way up.

However integration testing for this project my opinion is different. An OSGi application consists of components that are packaged in a bundle. That means that if some bundle isn't available on the OSGi framework, the application cannot be started. This means that we need to be able test the life cycle of these bundles. Another problem is that these frameworks can work together in clusters. If one of these frameworks shuts down what will happen with that cluster? Testing the life cycle of the framework is also a must. These two things formed a definition for integration testing in this project. The definition of integration testing used in this project is: "*Be able to test the life cycle of bundles and the life cycle of the targets in an environment that is like the production environment*".

OSGI

OSGi is dynamic module and service platform for java which allows components/application to start dynamically without restarting Java VM. A OSGi framework divided in 5 different layers.

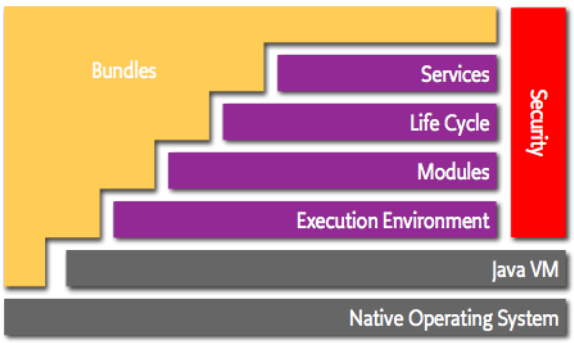

Figure 4 OSGI layers http://www.osgi.org/wiki/uploads/About/layering-osgi.png

Execution environment : This is where OSGi can run on. It can be a Java VM.

Modules: *"Modules layer defines the class loading policies. The OSGi Framework is a powerful and rigidly specified class-loading model. It is based on top of Java but adds modularization. In Java, there is normally a single classpath that contains all the classes and resources. The OSGi Modules layer adds private classes for a module as well as controlled linking between modules. The module layer is fully integrated with the security architecture, enabling the option to deploy closed systems, walled gardens, or completely user managed systems at the discretion of the manufacturer."* http://www.osgi.org/About/Technology

Life Cycle: *" Life Cycle layer adds bundles that can be dynamically installed, started, stopped, updated and uninstalled. Bundles rely on the module layer for class loading but add an API to manage the modules in run time. The lifecycle layer introduces dynamics that are normally not part of an application. Extensive dependency mechanisms are used to assure the correct operation of the environment. Life cycle operations are fully protected with the security architecture, making it virtually impossible to be attacked by viruses. "* http://www.osgi.org/About/Technology

Services: *"The service registry provides a cooperation model for bundles that takes the dynamics into account. Bundles can cooperate via traditional class sharing but class sharing is not very compatible with dynamically installing and uninstalling code. The service registry provides a comprehensive model to share objects between bundles. A number of events are defined to handle the coming and going of services. Services are just Java objects that can represent anything. Many services are server-like objects, like an HTTP server, while other services represent an object in the real world, for example a Bluetooth phone that is nearby."* http://www.osgi.org/About/Technology

Security Layer: *"Security is based on Java and the Java 2 security model. The language by design limits many possible constructs. For example, buffer overflows used in viruses are impossible. Access modifiers in the language restrict the visibility of the code to other programmers. The OSGi platform extends this model by allowing private classes, a mechanism that is not available in a standard way in Java. The Java 2 security model provides a comprehensive model to check access by code to resources. The OSGi platform adds full dynamic management of the permissions, simplifying the life of operators and system administrators."* http://www.osgi.org/About/Technology

DISTRIBUTIONS OF TESTS

In order to get the tests on multiple targets you need to be able to distribute the tests to the targets. A lot of members of Luminis Technologies are currently developing on a OSGi provisioning system called Apache ACE which allows applications to distributed to the containers that are managed by Apache ACE. Eclipse uses a different provisioning system to install and update new software. This system is called Equinox P2 and it also allows one to deploy to OSGi containers. Both systems have a similar workflow of provisioning applications. We are going to compare the two systems to see what the systems are capable of. The features that systems need to be capable of doing are:

- Be able to distributed to multiple targets
- Be able to change what is deployed on the targets

The systems were compared and the results are:

Feature	Apache ACE	Equinox P2
Distribute to multiple targets	x	x
Be able to change what is deployed on the target	x	x

Both systems are meeting the two features that are required. So both can be used in the project? Based on the comparison yes, but because the members of Luminis Technologies are working on and with Apache ACE this system will be used for this assignment.

HOW SHOULD ACE BE IMPLEMENTED IN THE PROJECT?

The most obvious way to use ACE is for its provisioning abilities. It allows us to distribute our test packages on different targets. The framework needs to communicate with ace for the distribution of tests. The test framework also need to communicate with the target for executing.

CURRENT TEST FRAMEWORKS

There are a couple of test frameworks that can test OSGi bundles. These are Pax Exam, BND integration testing tool and Arquillian. We are going analyze these test frameworks. The test frameworks need to meet a set of requirements.

These requirements are :

- It needs to be easy to write a test.
- You should be able to test on a existing target.
- You shouldn't spend more time on configuring the test framework.
- Test classes and bundles that are going to be tested are deployed in separated bundles.
- You need to be able define additional bundles that need to be deployed.
- Must be able to test on multiple targets.

For explanation of each test framework please check the research document. The test frameworks were compared and analyzed with the requirements.

Requirement	Pax Exam	BND Integration testing Tool	Arquillian
It needs to be easy to write a test	+	+	+
You should be able to test on a existing target.	-	-	+
You shouldn't spend more time on configuring the test framework.	-	+	-
Test classes and bundles that are going to be tested are deployed in separated bundles.	+	+	-
You need to be able define additional bundles that needs to be deployed.	+	+	+
Must be able to test on multiple targets.	+	-	+

I've chosen to use Arquillian based on that it supports testing on existing targets and multiple targets. BND Integration testing tool can only test on a target that the framework has created. It doesn't support to test multiple targets because you can't define this. Pax Exam does support multiple target testing however it isn't very user friendly because you need to write a lot of code for the configuration of target than writing the test . Arquillian takes some time to configure it and that is unable to deploy the bundles and test separately. Because is Arquillian is an extensible framework we can solve those problems.

The next part of the research was to design the test framework in order to do this. For this, the working of Arquillian needed to be researched because there were some questions when comparing the test framework:

- How does Arquillian know what to deploy?
- How does Arquillian know where to deploy?
- How does Arquillian communicate with a container?
- How is Arquillian able to intercept the JUnit life cycle to insert the test result?
- How does the implementation of the OSGi container work?

5.3 CONCLUSION

The main purpose of the research was to analyze the possibility of integration testing of OSGi applications that are distributed over multiple nodes by researching and comparing existing solutions. It is possible to do this by creating a test framework that combines a couple of technologies. The test framework will handle the distribution of the tests and required resources with the help of Apache ACE. The reason to use Apache ACE is for its provisioning features of OSGi bundles and not Equinox P2 is not based on features because they can do more or less the same thing. The choice was preference, because Apache ACE is developed by a couple of members of Luminis Technologies and they are using it for their development. It is good thing to also use ACE because the members are familiar with ACE.

Arquillian is used for the deployment to Apache ACE and for the executing the targets. The research compared three test frameworks that can test OSGi bundles. These were Pax Exam, BND integration testing tool and Arquillian. All three were analyzed by with a couple requirements. The reason to choose Arquillian above the others is that Arquillian is able to test on existing targets and the other frameworks only could test on targets that they have created. But in order to use Arquillian things needs to be changed. A new implementation needs to be build to support Apache ACE. A wrapper for ShrinkWrap needs to be build in order to deploy additional bundles to the target. How to the test framework looks like is found in "How does the test framework look like?"

5.4 HOW DOES THE TEST FRAMEWORK LOOK LIKE?

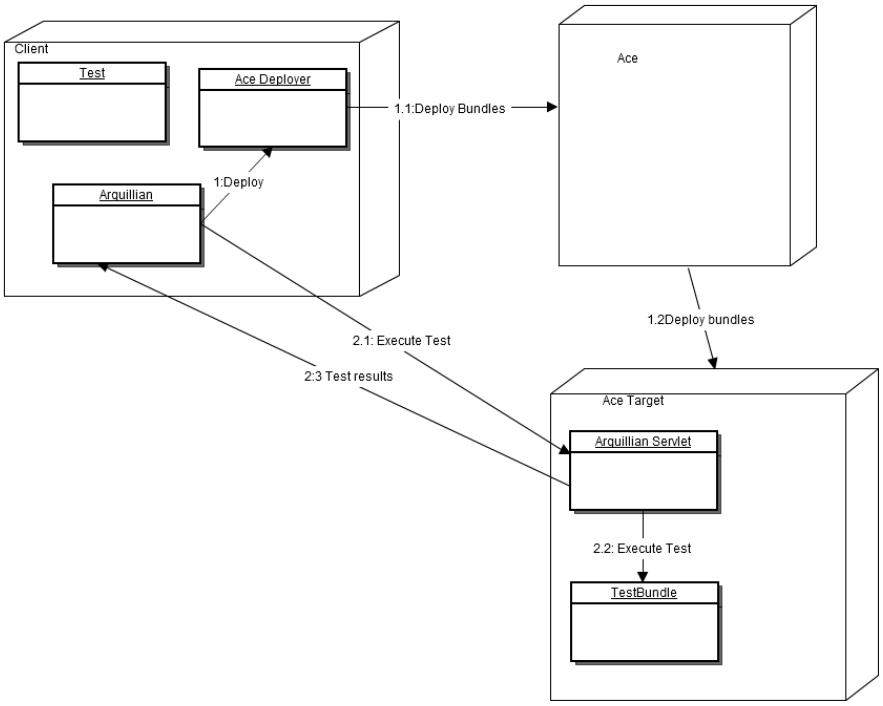

Figure 5 Initial Design

The test framework consist in a couple parts that are running on different location. These parts are:

- Deployment
- Distributing
- Execution

DEPLOYMENT

For the deployment of the tests we are going to use Arquillian. But we need to build an implementation that communicates with Apache ACE. The implementation is the Apache ACE Container. It will use the web service API for the deployments. Arquillian doesn't allows us to deploy the test packages and the required resources in separated bundles. We can solve this problem by creating a Wrapper for ShrinkWrap that allows us to define what bundles need to be deployed and deploys this before that test is executed on the target.

DISTRIBUTION

Apache ACE is going to be used to distribute the bundles and tests to the targets. In order to get the things on the target ACE needs to know where and what to distribute. These instructions are given when the ACE Deployer deploys the tests and resources to ACE.

EXECUTION

Executing is a technical part. For the user that want to know the technical overview of the test framework please check the architectural overview document and for a summary please check the chapter "How does the test framework look like?" in the research document.

To execute a test you need a couple of things. First of all ACE should be running and there needs to be a target that ACE can deploy to. Secondly a user needs to include the test framework in the project and need to configure the locations of ACE and the target in the arquillian.xml

```
<?xml version="1.0"?>

<arquillian xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xmlns="http://jboss.org/schema/arquillian"

    xsi:schemaLocation="http://jboss.org/schema/arquillian
http://jboss.org/schema/arquillian/arquillian_1_0.xsd">

    <container qualifier="Ace" default="true">

        <configuration>

            <property name="feature">feature</property>

            <property name="distribution">distribution</property>

            <property name="target">server3</property>

            <property name="targethost">localhost</property>

            <property name="host">localhost</property>

            <property name="port">8080</property>

        </configuration>

    </container>

</arquillian>
```

This is an example of a Arquillian.xml file . The properties that are defined are going to be used to locate the targets and ACE server.

The next thing is to create a test. Below is how a simple test is written. For the test three things are required:

- @RunWith(Arquillian.class)
- @Deployment

```
@RunWith(Arquillian.class)

public class TestSuite {
    public test1() {
    }
    @Deployment
    public static JavaArchive createdeployment() {
        return ShrinkWrap.create(JavaArchive.class);
    }
    @Test
    public void  test1() {
        assertTrue(true);
    }
}
```

  public static JavaArchive createdeployment()
- @Test method

This is an example how to write a test in order to get the test framework working.

To test this on multiple targets we need to use a other way to test.

```
Public  void testStuff() {

        Target t = AceDeployer.getTarget("server1");
        t.execute(TestSuite.class);
        Target t2 = AceDeployer.getTarget("server2");
        t2.execute(TestSuite.class);
        }
```

The code explains how to get the information of a target. The code would then use the information to execute the test on that target. The test that is going to be executed in the example code for a simple test.

6. PROCESS

This chapter explains the process of the project. Because we worked with SCRUM the process was iterative meaning that phases could be done parallel. The kickoff explains how the project is started. The requirements paragraph explains how the requirements were retrieved. "Every day is a new day" explains what the work process was on a daily based. Design and Implementation describe the process of the design and implementation.

6.1 THE KICKOFF

The kickoff started at the interviews before that I started with the graduation project. There were three assignments and the choice was this project. Back then it was still unclear what they expected me to do. The assignment that I've chose was using technologies that were unknown to me. This was a problem because if you know don't know what technologies do, how can you understand the assignment? When the graduation finally started, the first thing that was on the agenda was the Project Initiation Document(PID). This document contains the a plan how this project should be approached and what are the risk that can occur. A time planning was made and it was included in that document. When the PID was done, I started a research plan. This plan is a way to get a grip on the research. What needed to be researched ? What was the problem? how am I going to research all this? All these questions were answered when writing the research.

6.2 REQUIREMENTS

The requirements for the test framework were already defined at the beginning of the project. With these requirement I could start a research to find test frameworks and distribution systems. However the requirements needed to be prioritized and to do this I asked the input of the users in the form of interviews. In these interviews questions were asked about the problem that the users had with testing and what the solution should be capable of. The answers were converted into requirements. The requirements combined with the ones in the beginning of the project were then prioritized and written in forms of user stories. User stories are stated as followed: "I want to be able to do something with something." This way you could see how the user want to be able to do something. With these user stories test scenarios were defined. These scenarios describes how the user should use the test framework and what the test framework should be able to do.

6.3 EVERY DAY IS A NEW DAY

Every day starts with a standup talking about my progress and the problems that I was dealing with. The thing about this project is that my users are involved in these standups. This is because one the user is the SCRUM master. The users aren’t satisfied quickly. For example when I solved a piece of the puzzle, I told them how I solved it and they always gave me feedback how to improve the solution. This can sometimes be frustrating because you worked hard on it. After couple of times you got used to it and you are going to expect feedback about how to improve the solution. The project can be explained in two parts. The first part is the research. The second one is the more communicative one where stuff needs to be discussed like requirements, software designs etc... In the next paragraphs these parts will be explained. Researching technologies and discussing this with the project was my daily work routine. In the beginning this was to get familiar with the technologies but this changed when the components for the test framework were chosen. My daily routine turned into implementing the components of the solution. After deciding to use Arquillian and Apache ACE for the test framework a design needed to be made.

6.4 DESIGN

To make the design we made an initial global design how the test framework should look like. This design was global and there were questions about how the communication went between the test framework and ACE and target. The questions were part of the research and were answered in the research. The design for the test framework changed based on the result of the research. For example Arquillian had an OSGi implementation. This implementation consisted out of two parts. This first part was running on the client side. The second on the target. The part on the target was creating a test runner. This test runner created an extra test cycle. This resulted in two test cycles. One on the client and one the target. In early designs this test runner was still there. But after discussing about how the tests should be executed on the target this changed. The change was to make it simpler to execute the tests on the target meaning we were going to strip the part that was running on the target.

The design changed every time when a problem was solved. These problems could be with Arquillian or with Apache ACE. The more problems were solved, the better the design got. The design was mainly captured in a sequence diagram that got changed after discussions. The sequence diagram explains how the flows goes between components of the test framework. This is useful when discussing the test framework with the team. The components of the framework are visualized in the components diagram. With this diagram we could discuss which components were running where. For design of each class and what is does was up to me. The concept of the working the test framework was always discussed with the users. For example we got a problem with loading the tests on the target. When the tests are deployed on the target and Arquillian gave a signal to the target to start the test. The target then needs to locate the test but it how should this work? How to solve this was discussed with the users.

This previous paragraphs were about the technical parts of the test framework. But there is also another part and that is the usage of the test framework. How do users are going to use the test framework? How do they write the test? These question were also asked in the discussions about the test framework. We would write some example code how the user would write the test. These discussion were about usability of the test framework. This resulted in how the test framework be used.

6.5 IMPLEMENTATION

When a problem was solved and a designed was completed, the next step was to implement it. In other words programming the code to get it working. A set of tools were required to do this. You need an environment to produce the code and an environment to test the code. The programming environment consists out of Netbeans and Eclipse. The reason for two programming environments was that a lot of Arquillian code is in maven projects. These projects are easier to use in Netbeans for the intern. Eclipse however is easier to use to programming OSGi project because it supports BndTools. BndTools makes it easier to configure the OSGi project by giving the developer a graphical user interface(GUI) and it would automate the creation of the bundle. The project is using Java technologies for all the produced code. The reason is that everything is built in Java.

The test environment was an Apache ACE environment: The ACE server and a target. In the beginning it took a lot manual work to get tests and parts of the test framework that were programmed on the test environment. The reason was that the components of the test framework still needed to build. The further the implementation went the more easier it was to get things on the test environment. When a part was implemented it was discussed. The feedback that came out of the discussions could have effect on two parts: the implementation or the design. Both ways affect the implementation because when the design changed the implementation of the design also changed. The implementation started with getting the OSGi implementation of the Arquillian working. The next step was to look how it was working and discussing this with the users.

Feedback was given about the implementation and how it should be working. With this feedback parts of the implementation were rewritten and replace the original parts.

Working this way I still had my working test environment and I could test if the part that is rewritten worked. The parts that were working were again discussed. The users gave the feedback to improve the part or to go on with the next part. When rewriting more parts, I came to the conclusion that there was nothing left of the OSGi implementation. Almost everything was replaced. The only parts that were kept are the core features of Arquillian. These were parts of the life cycle of Arquillian. Things that were newly created were parts to communicate to ACE and parts to make users to configure the deployment.

The code that the intern had produced is managed by a version system called Git and is hosted on Bitbucket. This results in that the code is always retrievable. When the test scenario: "*To successfully run a test on one target*". A release was made so that the users could try it. It took about 2/3 weeks to get this release working. The main reason for this was that the intern was using another operating system (OS) than the users. The whole Java file I/O is different. This first couple of times the intern blindly released the code without knowing this. Later releases were tested on similar operating systems as the users before releasing it to the users. If the users got any exceptions with the test framework, it was my jobs to find out what the problems were. The users described what the problem was, and the steps they took to reproduce it. The first couple of problems were distribution problems. Because of the different operation systems I couldn't reproduce the problems. I didn't know that File I/O in java was different. The extra step before releasing it to the users was to test it in another OS.

6.6 RESULTS

The result of the project was a test framework that could test the distribution of the bundles to multiple targets. The test framework is able to deploy and undeploy the bundles. However I couldn't make the test framework to test the life cycle of the targets. Another thing that I was unable to do was to be able to tests on multiple targets at the time of writing this paper. However I still got a couple of weeks before the presentation to build the feature in the test framework. So that is advice for improving the test framework. I wrote a couple of documents that support the test framework. These documents are architectural overview of the test framework, research of this project and requirement document. At the end of the project a presentation will be given so that the users can see how the test framework can be used.

The actual planning was :

Weeknr	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
PID																	
Research plan																	
Requirements Document																	
Research answers																	
Functional Design																	
Technical Design																	
Research Results																	
Prototype																	
Test Document																	
Documentation																	
Graduation paper																	

7. CONCLUSION

The project was about finding a possible way to test the integration of OSGi projects that distributed over multiple nodes. A couple of current test frameworks and provisioning systems were compared for their features with the requirements of the users. The comparison for the provisioning system resulted in choosing Apache ACE. The reason for choosing ACE was based not based on the features, because the other system had the same features. The decision for choosing Apache ACE was that the members of Luminis Technologies were using the Apache ACE and are familiar with the system. The comparison of the test framework resulted in choosing Arquillian. Arquillian was chosen for the its ability to test on existing targets and for its extensibility. These two components are part the test framework that the developer can use to test the integration of OSGi Projects in multiple nodes.

7.1 ADVICE

The test framework isn't finished yet. There are a couple of issues with the test framework. The most important issue is that the test framework is unable to create a target to deploy the tests to. This is future work.

8. RETROSPECTIVE

I'm looking back to this project in two ways. This first way as a project member and the other way is my personal experience. The iterative way of working suited this project really good. The reason that it worked great was the type of the project. In the beginning everything was new to me and the further the project went the more I saw the project progress. In the beginning the components were just components not working together. Step by step these components were working together. My mentors/users did a great job, without their input of how the test framework should look like, I could not have built it. The daily standup worked really great. That extra communication every day made this project a success. You can basically say the project was managed each and every day. The feedback could be sometimes a bit harsh after you spent a lot of time making something to work. But it was that feedback that was key to the project. Communication went really well but could be improved. There were times where we weren't on the same page with the design. Again with the daily standup we could get back to the same page. Another thing that could go better was the delivery and planning on Fridays. On Fridays, members of the Leren-op-maat project team come together for their release. The same day that I got my offline meeting with my mentors. My mentors are part of the Leren-Op-Maat team and they are busy with their release. Their time is precious and I sometimes thought not to bother them, which could delay the project.

My experiences as an intern at Luminis Technologies were great. I've learned a lot. For example my purpose of this project was to improve my communication skills. In my opinion I achieved a part of my goal. In the beginning when I explained what my problem was with for example the Arquillian code. When I tried to explain a design or a problem they had a hard time understanding me. The reason for this is that I tend to start talking about details directly, without introducing what I am talking about. I practiced on improving this during the project by explaining the design in more detail over and over again. I've learned to improve this. Another great skill that my mentors taught me was to visualize the sequence of the applications with real people. Each person was a component of the application and we then communicate what data we passed to the person. It's a sort 'role playing'.

I've worked most of the time at home. This was a new experience for me. Working at home helped me a lot because to go to the office every day took me 2.5 hours. When I worked at home my productivity increased. On a day I could do much more at home than when I went to the office. The communication over the internet went fine. With Google Hangout I could share my screen with the others. This assisted me with my explanation of my problems. Sometimes I miss the real interaction that you get in the office. When communicating with people you are always watching how are they reacting. This is something that I missed with the online meetings. We used webcams but that wasn't enough for me to tell really how they would react to me.

With the end near I could sum my experience at Luminis Technologies as a great experience!

REFERENCES

BOOKS

Verschuren, Piet & Hans Doorewaard (2007) *Het ontwerpen van een onderzoek*.(4e dr) Uitgeverij LEMMA

Fredriksz, H., Hedeman, B. & Heemst, G.B. van (2009). *Projectmanagement op basis van PRINCE2 - Edition 2005* (3e dr),Van Haren Publishing

WEBSITES

Last checked on 4-06-2012

Amdatu: http://www.amdatu.org/confluence/display/Amdatu/Welcome+to+Amdatu

Luminis : http://www.luminis.eu/en/expertise/software-craftsmanship/

Luminis Technologies: http://luminis-technologies.com/

Apache Foundation: http://www.apache.org/foundation/

[tsp] Tien Stappen Plan : https://fhict.fontys.nl/IS/afstuderen/Lesmateriaal/Korte%20samenvatting%20TSP%20voor%20Afstudeerders%20met%20een%20Ontwikkelopdracht.doc

OSGi: http://www.osgi.org/About/Technology

MoSCoW: http://www.ietf.org/rfc/rfc2119.txt

Integration testing : http://searchsoftwarequality.techtarget.com/definition/integration-testing

http://www.freetutes.com/systemanalysis/sa9-top-down-integration.html

http://www.freetutes.com/systemanalysis/sa9-bottom-up-integration.html

Apache ACE: http://ace.apache.org/

Equinox p2: http://www.eclipse.org/equinox/p2/

http://eclipsesource.com/en/downloads/presentations/remote-provisioning-with-p2/

Pax Exam: http://www.javabeat.net/2011/11/how-to-test-osgi-applications/

Arquillian: http://arquillian.org/invasion/

ShrinkWrap: http://www.jboss.org/shrinkwrap

ATTACHMENT I PROJECT INITIATION DOCUMENT

ATTACHMENT II RESEARCH DOCUMENT

ATTACHMENT III REQUIREMENTS DOCUMENT

ATTACHMENT IV ARCHITECTURAL OVERVIEW