



Java Bluetooth stack Project Description

Version 1.0

Revision History

Date	Version	Description	Author
2003-11-12	0.1	Initial Draft	Tomislav Sečen
2003-11-17	0.15	Minor changes	Tomislav Sečen
2003-11-18	0.17	More minor changes	Tomislav Sečen
2004-01-10	1.0	Final version	Tomislav Sečen

Table of Contents

1.	Introduction	5
1.1	Purpose of this document	5
1.2	Intended Audience	5
1.3	Scope	5
1.4	Definitions and acronyms	5
1.4.1	Definitions	5
1.4.2	Acronyms and abbreviations	5
1.5	References	6
2.	Organization	6
2.1	Project Manager	6
2.2	Project group	6
2.3	Supervisor	6
2.4	Customer	6
3.	Assumptions and constraints	6
3.1	Technological	6
3.2	Environmental	6
3.3	Interpersonal	6
3.4	Time	6
4.	Deliverables	6
5.	Inputs	7
6.	Project risks	7
7.	Communication	7
7.1	Project language	7
7.2	CVS	7
7.3	Web site	7
7.4	E-mail	7
7.5	Scheduled meetings	7
8.	Configuration management	8
8.1	Responsibility and Resources	8
8.2	Components, Tools and Products Used in the Project	8
9.	Project plan	9
9.1	Time schedule	9
9.2	Activity plan	9

1. Introduction

To utilize any Bluetooth device, one needs a pretty complex stack of protocol layers, much like the OSI referent model. With the Bluetooth device you buy in the store, you get that stack implemented by the manufacturer and you install it as a 'driver' for your newly bought hardware. The problem is that this 'driver' (which is in fact a Bluetooth stack) doesn't expose any developer-oriented functionality except Bluetooth profiles (which ones? – it varies from one manufacturer to another) on virtual COM ports, so developer is constrained to use that COM ports in his Bluetooth application, without the possibility to harvest advanced Bluetooth features (at least not programmatically – user can only do it manually) like device and service discovery of devices in the vicinity.

There are many Bluetooth stacks out there, mostly written in C (like BlueZ for Linux) or in ASM for some microcontroller. There are even some Java Bluetooth stacks, but there are either commercial products (like the one from Atinav or Rococo) or left in some undetermined state of development (like Harald Bluetooth stack).

The task presented here is to:

1. Examine the existing stacks that are out there.
2. Create a fully compliant JSR-82 stack, upgrading existing code.
3. Publish it on sourceforge.net!

1.1 Purpose of this document

Purpose of this document is to give a project plan, provide development guidelines, work division and resource management for the Java Bluetooth stack project.

1.2 Intended Audience

Product supervisor, development team.

1.3 Scope

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Java	The programming language
Bluetooth	Wireless communication technology
Eclipse	Open-source development environment, formerly IBM WebSphere

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
J2ME	Java 2 Micro Edition
UML	Unified Modeling Language
JABWT	Java APIs for Bluetooth wireless technology
CLDC	Connected limited Device Configuration
JSR-82	Java Specification Request 000082 – Java APIs for Bluetooth wireless technology

1.5 References

2. Organization

2.1 Project Manager

Tomislav Sečen

2.2 Project group

Name	Responsibility
Marko Đurić	Testing/Documentation
Sanjin Goglia	Implementation/Integration
Dražen Njerš	Implementation/Integration
Tomislav Sečen	Project Manager/SCM/Design/Integration

2.3 Supervisor

Igor Čavrak igor.cavrak@fer.hr

2.4 Customer

Igor Čavrak igor.cavrak@fer.hr

3. Assumptions and constraints

3.1 Technological

- Java is used as the main development language.
- The CVS Server must be up and running "all the time".

3.2 Environmental

- All necessary development software must be installed on the machines.

3.3 Interpersonal

- We are limited by the knowledge in our group. It is not possible to hire external personnel.

3.4 Time

- The project must be completed by week 02, year 2k4.

4. Deliverables

To	Output	Planned week	Delivered week	Rem
Project web	Project description	46	46	
Project web	Requirements definition	48	49	
Project web	Project design	49	47	
Supervisor	Test plan document	2	2	
Customer	Final product delivery (presentation, source code, user documentation)	3	3	
Project web	Final project report	2	2	

Comment: *Planned week* = week when it is required by the project; *Received week* is week when it was actually received; *Rem* is a remark index number.

5. Inputs

There are no 'hard coded' inputs; inputs will be useful tips from project supervisor and guidelines from the customer.

6. Project risks

Possibility	Risk	Preventive action
High	Someone (competition) releases their JSR-82 compliant stack	Make our project even better and more complete
High	Progress gets late	Constantly check the progress and take action as soon as there are signs of missing deadlines – reduce demands
Low	A project member drops out	The other members take over.
Low	The CVS Server gets hacked	Keep project copies on developers' machines

7. Communication

7.1 Project language

The project language is English. All documentation will be written in English.

7.2 CVS

All documents will be published on the project web page. These documents are available for all the group members and for the supervisor.

7.3 Web site

On DSD web page (<http://www.fer.hr/rasip/dsd/>) there will be a project folder (Projects>Bluetooth). It will include:

- Important documents about the technologies involved in the project development.
- Information about all group members (hopefully?).
- News & information showing when meetings are planned and the project achievements together with other useful information that can occur.
- All documents produced which are public to anyone.
- External resources like links to useful information on the Internet.
- Guidelines for documentation, coding and CVS.

7.4 E-mail

We will use emails to pass information around quickly to the group members. There will be a weekly progress report delivered via email to project supervisor every Friday morning.

7.5 Scheduled meetings

To keep a good communication we will have to have frequent meetings where we compare the work with the timetable and discuss common problems. To avoid problems we will try to work together some times a week. On the meetings we also discuss the work hours made a week to make sure that everyone does their work.

Project meetings with the supervisor will be held whenever necessary.

Time for the weekly group meeting(s) is Tuesday at 1100 or Thursday at 1600.

8. Configuration management

8.1 Responsibility and Resources

Project leader will take on the SCM tasks.

SCM tasks:

- Write instructions how to use CVS from Eclipse development environment.
- Create and manage Eclipse projects
- Check that everyone in the project group is using CVS in the right way.
- Create the final product from the software library.

8.2 Components, Tools and Products Used in the Project

The following SCM tools are used in the project:

- Eclipse 2.1.2
- JCraft – an Eclipse SSH2 plugin for extssh2 connection method

The following development platforms are used in the project:

- Windows NT/XP/2000
- Java SDK 1.4.2
- Eclipse 2.1.2 (prev. IBM WebSphere)

The following tools are used in the project:

- EclipseUML plugin from Omondo
- Microsoft Word & PowerPoint
- Microsoft Internet Explorer 5 (or higher)

