Software Development Plan - JMSME

Paweł Ostrowski

15 czerwca 2003

1 Introduction

1.1 Purpose

This document is a realization plan of the JMSME project. It discusses all realization stages, methods of project organization and project management.

1.2 Scope

This document concerns JMSME (Java Messaging Service Micro Edition) project. It specifies the work plan, which consists of:

- project description
- project organization
- project management
- technical project plans
- auxiliary plans

1.3 Definitions

JMSME - The name of the project, stands for Java Messaging Service Micro Edition

Open Source - as defined on www.opensource.org

J2ME - Java 2 Micro Edition

J2EE - Java 2 Enterprise Edition

JMS - Java Message Service

J2SDK - Java 2 Software Developers Kit

Client-side - JMS implementation classes running on J2ME device

Server-side - JMS proxy running on J2EE middle ware machine

MIMUW - Faculty of Mathematics, Informatics and Mechanics Warsaw University

ZPP - Team Programming Project, part of curriculum at MIMUW

1.4 Attachments

1.5 Document Summary

Further document chapters deal with all plan aspects.

2 Project Discussion

2.1 Project Goals

The main project goal is to develop open source, functional JMS implementation for J2ME platform. For more concepts and further discussion see Vision JMSME.

2.2 Assumptions and Constraints

2.2.1 Budget

The JMSME project is going to develop a free software product, and it is meant to be realized without budget, just with hard work of JMSME team.
At all work levels the project is done using free software. These include:

CVS - for synchronizing team work on documentation and code

LaTeX - for creating all documents

Poseidon for UML, community edition - or other program of similar functionality, for creating UML diagrams

J2ME - for running client side

J2EE - for ruining server side

J2SDK - for creating both client and server side

2.2.2 Staff

This plan assumes work of four person JMSME team, which does all the project work.

2.2.3 Equipment

Development process is carried using team's own home computers, and computers from MIMUW's laboratories. The CVS repository is located on lucash's private machine.

Product testing requires J2ME devices. It is going to be done using software emulators and/or borrowed mobile devices.

2.2.4 Schedule

Name	Release date	Description
Library alpha version	23.04.2003	Working proxy and client classes, perhaps lacking some functionality
Ready system	04.06.2003	Working proxy and client side providing full functionality of JMSME system. Full integration with JMS providers server.

In the following chart 'Client' means client-side part of JMSME system, 'Proxy' means server-side part of JMSME system and 'System' means classes common for 'Client' and 'Proxy'.

All 'Already done' implementations listed below are test implementations, propably unstable and subject to change. All classes are written along with corresponding test classes where possible.

Date	**Description**
Already done	**Client:** • Implementation of all *Message interfaces from JMS api (classes: Message, TextMessage, BytesMessage, MapMessage, StreamMessage from jmsme.client.jmsimplementation). • TRLControl and TransputerListFactory implemented. • Simplified version of JNDI framework. It can now make lookups for ConnectionFactories. • Enriched TransputerList for client-side. • MessageProducer, MessageConsumer, Session interfaces implemented. • Abstract storage and network implementation. • Implementation of network transputer on http protocol. **System:** • Implementation of internal TransputerList framework (TransputerList, Transputer) • JMSHashtable implementation with getXXX methods, streamable. • OperationObject framework (classes: OperationObject, OOF).

Date	Description
02.IV.2003	**Client:** • First implementation of Destination, ConnectionFactory, Connection classes. • Bug hunting. Further development of 'already done'. • MIDP implementation of storage transputer. **Proxy:** • Connection initialization support. • Http communication support.
9.IV.2003	**Client:** • Transaction and UDP network transputer implementation. • Unit tests of all written client-side classes. **Proxy:** • UDP based support. • Framework for translating OperationObjects into JMS API calls. • Unit tests of all written proxy classes.
16.IV.2003	**Client:** • Support for automatic sending of pending messages. First module tests, first attempts of make it all work together. First beta version should appear this week.
23.IV.2003	• Module and Integration tests continued. • Intensive debugging. • Code cleanup. • We will put the project on the Source Forge this week.
30.IV.2003	Break
May	All things we did not manage to do in April.

2.3 Project Artifacts

Project work is divided into two stages. First one (from October 2002, to January 2003) yields the following documents:

- Vision 06.XI.2002
- Business Use Case 20.XI.2002
- Software Development Plan 20.XII.2002
- Use Case (first version) 20.XII.2002
- Use Case (final version) 15.I.2003
- Software Architecture Document 15.I.2003

The second work stage is implementing JMSME products

- client-side classes
- server-side proxy server

2.4 Version Change Process

Release Date	*Document Version*
20.XII.2002	First Version
ca. 15.I.2003	Revised First Version
ca. 20.II.2003	Second Version covering Implementation Phase
ca. 7.III.2003	Revised Second Version

3 Project Organization

3.1 Organizational Structure

The JMSME team consists of four people:

- Paweł Ostrowski - project manager
- Łukasz Osipiuk
- Paulina Kania
- Piotr Anders

Assistance and support is provided by Grzegorz Grudziński who is Project Supervisor. He controls project requirements, and reviews work results.

Team members use nicknames internally in project work (eg. document change logs). This chart provides reference:

Nickname	**Person**
pasza	Paweł Ostrowski
ofca	Paulina Kania
lucash	Łukasz Osipiuk
piotrek	Piotr Anders

3.2 External Contacts

JMSME team will contact Project Supervisor regularly for project guidance and reviewing of work results.

JMSME team will also meet other ZPP teams and Project Supervisor every two weeks and attend technology presentations (see 4.2.3).

3.3 Responsibilities

This chart gives some idea about each JMSME team member's role.

Role	**Person**	**Responsibilities**
Project Manager	Paweł Ostrowski	Every normal project developer responsibilities plus: Team work's coordination. Contact with Project Supervisor. Successful project completion assurance. Controlling work advance.
Project Developer	Łukasz Osipiuk	Creative work, giving some concepts, everything to help successful project completion
Project Developer	Paulina Kania	Creative work, giving some concepts, everything to help successful project completion
Project Developer	Piotr Anders	Creative work, giving some concepts, everything to help successful project completion

4 Project Management

4.1 Estimates

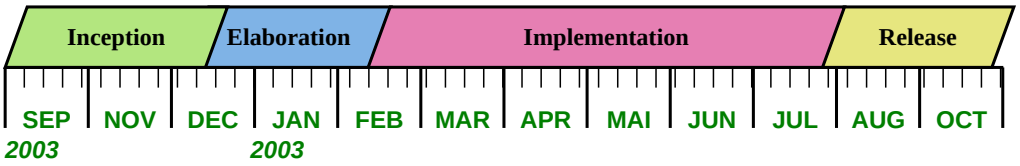
- **Budget** - Project is realized with no budget, any needed resources are provided by MIMUW or JMSME team members.

- **Schedule** - Project is realized between October 2002 and June 2003

Re-estimation may occur after project completion in June 2003. Project development will be continued if the project results are good enough.

4.2 Project Plan

4.2.1 Project Phase Plan



Phase	TODOs	Results
Inception	work planning, gathering potential requirements, gathered requirements analysis, use case analysis	Project scope definition, creation of the following documents: Vision, Software Development Plan, Business Use Case
Elaboration	software architecture planning	creation of the following documents: Software Architecture Document
Implementation	Writing the JMSME code, testing the software, porting existing open source JMS application to JMSME	creation of JMSME products, creation of demo application for JMSME
Release	Presentation of project capabilities, project release, putting project at sourceforge	making products available

4.2.2 Project Schedule

The first part of project schedule includes creation of the following documents:

Document	Date
Vision	06.XI.2002
Business Use Case	20.XI.2002
Software Development Plan	18.XII.2002
Use Case (first version)	18.XII.2002
Software Architecture Document	15.I.2003
Use Case (final version)	15.I.2003

Further project schedule is not specified. It will be noted here when specified.

4.2.3 Training Plan

ZPP Teams will prepare presentations to introduce some technologies concerning all ZPP projects in common.

In Inception project phase:

Presentation Topic	Date
LaTeX	23.X.2002
CVS	23.X.2002
JMS I	06.XI.2002
JMS II	20.XI.2002
Java	04.XII.2002
JavaDoc	04.XII.2002

There will be further, currently unspecified peresentations carried out in Implemetation phase. They will be noted here as soon as they are specified.

4.3 Project Supervision and Controlling

4.3.1 Requirements Management Plan

Requirements are divided according to their level of importance:

- **critical requirements** - Requirements which realization is essential for the project.
- **supported requirements** - Requirements important, but may be omitted for a reasonable purpose.
- **optional requirements** - Requirements which realization is not very important. They are satisfied only if it does not interfere with realization of requirements from other groups and it comes with a little effort.

4.3.2 Schedule Management Plan

Person responsible for realization of each project stage on schedule is Project Manager. He will ensure that each project work is taken with enough advance to be finished on time.

If it should happen that any work may be delayed, he will do all efforts to analyze the cause of delay and to reasonable extend eliminate it. If it should be impossible to omit work delay, the Project Manager will try to renegotiate the schedule with Project Supervisor.

4.3.3 Quality Control Plan

Project Supervisor is expected to review and verify all documents and other work results. All documents must be introduced to him and accepted before the final version.

Areas which are reviewed and must be accepted are:

- project decisions made in a document
- document style and comprehensiveness
- document language

If document is not accepted it is returned to author(s) and must be corrected.

Beyond this routine, Quality Control Plan encourages all team members to check work results of each other and to give suggestions/make corrections where necessary.

4.3.4 Report Plan

Reports will be prepared on demand of Project Supervisor.

Reports will discuss project progress.

4.4 Risk Management Plan

Description	Effects	Predicting	Avoiding	When it happens
Team's member leaving	Delay in project work	People who are more likely to leave: • people who do not come for ZPP meetings regularly • people who do not do their work	Controlling and motivating each other	Sharing missing person work, possible schedule renegotiation

Description	Effects	Predicting	Avoiding	When it happens
Temporary team member absence (eg. illness, personal problems)	Delay in project work	Team members stay in touch so prediction of such problems should be possible in advance	Project manager reminds all team members their responsibilities	Sharing missing person's work, possible schedule renegotiation
Delay in project work.	Bad work review, further delay possible	Appearing of a difficult, persistent problem may cause delay in project work.	Project manager checks all work advance regularly and assigns more people where necessary	Project schedule renegotiation possible, all night work :)

4.5 Project Completion Plan

Project Completion will include:

- Printing all the documentation
- Preparing repository archive and delivering it to all project developers
- Delivering all the documentation to Project Supervisor
- Presenting final product to public audience

5 Technical Process Plans

5.1 Used Methods, Tools and Technologies

Technologies used:

- Rational Unified Process - partial usage for business and technical modeling
- UML - usage for process modeling

5.2 Product Acceptance Plan

1. Final product presentation (using demo application)
2. Product is accepted by Project Supervisor

Product Acceptance Plan is prone to change in Second Version of this document.

6 Auxiliary Plans

6.1 Changes Management Plan

- Each project change will be immediately noted in appropriate documents.
- Using CVS helps keeping track of changes of documents and source code (each document has change log).
- In case of any problems with applying changes, each team member may easily contact project manager and other team members for discussion.

6.2 Project Evaluation

1. Testing all project modules by team members
2. Testing ready product by any volunteers
3. Complete project is going to be presented to some of the MIMUW's staff and public audience. They will provide some evaluation.
4. The most interesting evaluation factor will be eventual released product popularity.
5. Eventual feedback from product users.

6.3 Documentation Plan

Documentation is created by all team members and it changes frequently - all revisions are available via CVS.

Document correctness supervision: Project Supervisor.

6.4 Quality Assurance Plan

- Each project product is verified by at least two members of JMSME team.
- Each project product and document must be accepted by Project Supervisor.
- All quality problems must be solved.
- Project is interesting what increases motivation for creating good quality product
- All team members are eager to develop good quality products :)
- Additional motivation is perspective of passing ZPP with positive mark :)

6.5 Problem Solution Plan

- Every team member must immediately inform project manager about any problem in project realization.
- If project manager is unable to solve the problem, he contacts project supervisor.
- If project manager is unavailable other team members contact project supervisor.
- Any problems may result in delay of project realization, good team members cooperation is vital.

6.6 Process Improvement Plan

All procedures are discussed between team members. In case of any doubts, team asks for help Project Supervisor.

7 Changelog

```
$Log: sdp.tex,v $
Revision 1.9  2003/03/30 23:05:33  ofca
Drobne literowki, poprawki estetyczne. Nie zawsze podobaja mi sie
konstrukcje skladniowe, ale nie wiem czy mam probowac poprawiac cy olac :

Revision 1.8  2003/03/30 22:35:42  pasza
* doc/sdp.tex: dopisaliśmy (w mękach) harmonogram prac.

Revision 1.7  2003/03/27 18:32:45  pasza
* doc/sdp.tex: Dodałem tabelkę z terminami release‘ów projektu.

Revision 1.6  2002/12/20 21:40:30  pasza
Dodalem do changeloga orginalnego autora (robmar), żeby wstyd nie było,
w ten spsób jesteśmy bogatsi o 2 wpisy w changelogu

Revision 1.5  2002/12/20 18:41:30  ofca
kilka drobiazgow

Revision 1.4  2002/12/20 18:28:53  piotrek
Poprawki i uzupelnienia.

Revision 1.3  2002/12/20 15:52:59  ofca
drobne poprawki ciapkow
```

```
Revision 1.2  2002/12/20 13:08:09  pasza
napisałem sdp, spellcheck

Revision 1.1.1.1  2001/10/30 17:49:15  robmar
zajecia, etc.
```