



FVOPS

Project report

Status: Draft version 0.1

Authors: Dmitry Kolesnikov

Lappeenranta University of Technology

Department of Information Technology

Telecommunication Laboratory

P.O.Box 20, FIN-53851 Lappeenranta

Lappeenranta, October 01



Version History

Author	Date	Version	Note
Dmitry Kolesnikov	10/10/01	0.1	FVOP specification & implementation results

Content

VERSION HISTORY	2
1 INTRODUCTION.....	3
1.1 JVOPS	4
1.2 FRED.....	4
2 FVOPS.....	5
2.1 FVOP GOAL.....	5
2.2 FVOPS SPECIFICATION & IMPLEMENTATION	6
2.2.1 <i>Results</i>	7
3 FVOPS EVOLUTION AND RELATED WORK.....	7
3.1 OBJECT-ORIENTED EVOLUTION	7
3.2 JVOPS EVOLUTION.....	8
REFERENCE	9



Acronyms

EFSM	Extended Finite State Machine
FRED	Framework Editor
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
JVOPS	Java Virtual Operating System
OO	Object-oriented
RAD	Rapid Application Design
UML	Unified Modeling Language

1 Introduction

The implementation of a software protocol is not trivial task. The protocols must be efficient, reliable and flexible. It is more convenient to use software tools to create a protocol's implementation. Tools decrease cost of a protocol, reduce the complexity of network software, and make it easier to extend or modify protocols. At present time, there were developed huge amount of technologies and approaches that allows software engineers to rapidly create communication protocols. One of the most popular is based on the Conduit+ framework [1].

An object-oriented technology has been developed recent years. It allows software to be rapidly developed, maintained, and reused. It is followed that object-oriented frameworks are widely used to implement product line architectures; the protocol development is not exception.

An object-oriented framework is a collection of classes that implements shared architecture and the common behavior of a family of applications. The documentation of the framework depicts informally the specialization interface of product and plays significant role in development process [2].

In the research project FVOPS significant weight has two technologies. The first is an object-oriented clone of Conduit+ framework JVOPS. The second is a tool guides and controls application programmer according to the conventions of a framework called FRED.



1.1 JVOPS

JVOPS is a blackbox data communications protocol framework written in Java. The framework is light and is based on Conduits+ memo published in [1]. JVOPS has been designed to be an environment with fundamental possibilities of Java-based application development in distributed reliable computing system. It allows protocol designers to rapidly create a stack of communication protocols in common manner and use object-oriented technologies.

The framework is made up of two sorts of objects: conduits and information chunks [1]. Conduits are used to represent protocols, multiplexes and other components, each layer of a protocol stack is implemented by one or more conduits. Information chunks model everything that flows through conduits.

JVOPS is both a development time tool and run-time environment for protocol implementations. The first gives possibility to implement layers of a protocol stack from conduits. The second allows applications to use developed protocol for communication purposes.

1.2 FRED

FRED is a prototype tool intended for generating programming environments for Java frameworks [3]. The tool guides and control application programmers in implementing applications according to the convention of the framework [4].

An object-oriented framework is a collection of classes that implements shared architecture and the common behavior of a family of applications. Application-specific components interact with common, reusable framework through the specialization interface that extends the framework's hot spot [5].

The architecture and hot spot of framework are usually depicted by design patterns, but some time it is not obvious task how to design the specialization interface. There was developed tool called FRED that assist and control process of specialization interface creation.

The FRED gives possibility to both a framework developer to depict the hot spot for the framework, and an application developer. A hot spot of framework is depicted as a specialization pattern. Specialization patterns should not be confused with design patterns. The specialization pattern is a specification of a recurring program structure, which can be instantiated in several contexts to get different kind of structures [4]. A specialization pattern



definition is based on the analysis of the framework source code, UML diagrams and its documentation [6].

The specialization interface is created by an application developer where the interface is an instance of specialization pattern that is generated by FRED.

2 FVOPS

In recent years, the development focus has been changed from the original protocol stack implementation to embedded and application specific designs. Instead of developing standardized protocol stacks software application specific communication must be able to designed and implemented by Application Frameworks with protocol based communication support. These frameworks should therefore include features for solving protocol-oriented problems as well as tools for simulation and testing of implementation.

An FRED and JVOPS integration project based on protocol developers needs has been started in Lappeenranta University of Technology. The main project's goal is development of protocol-oriented application framework forRAD.

The project's duration more then one year it has been initiated at 01.05.2001. FVOPS results will be delivered after 31.12.2002 and they will be freely available. FVOPS includes four tasks that presented in the Table 1.

Task name	Mm	Period
FVOPS specification & implementation	9	01.05.2001 – 31.12.2001
FVOPS testing	6	01.01.2002 – 30.04.2002
Implementation of authentication and service selection applications	8	01.05.2002 – 30.12.2002
Project management	2	01.05.2001 – 31.12.2002

Table 1: Time schedule

2.1 FVOP Goal

It was mentioned above, the research project FVOPS is based on integration of FRED and JVOPS environment. The FRED supports systematic protocol development and the protocol software is created in JVOPS class instances.



An expected result is a Protocol-Oriented Application Framework that considers disadvantages of JVOPS environment. The Application Framework will cover protocol-oriented problems: simulation and testing of implementation.

The minor goal of FVOPS is further JVOPS development to direction that are required by RAD processes in telecommunication systems.

2.2 FVOPS specification & implementation

JVOPS provides Conduit+ framework that allows protocols to be implemented in object-oriented manner. A stack of a communication protocols is created from conduits that communicates by information chunk. Conduit+ has specified four conduits: Protocol, Mux, Adapter and ConduitFactory. These are hot spot of JVOPS.

The process of the specialization interface creation for JVOPS is required time resources and high developers' experiences, but FRED tool can minimize these requirements if specialization patterns exist for listed conduits.

The task *FVOPS specification and implementation* is aimed to create specialization patterns for conduits (Protocol, Mux, Adapter and ConduitFactory) and it includes wide collection of activities (See Table 2) that helps to perform JVOPS & FRED integration process.

Activities	Status
Conduit+ analysis	Done
JVOPS analysis	Done
FRED notation analysis	Done
Protocol Creator analysis	Done
JVOPS specialization pattern specification	In progress
- Protocol pattern specification	Done
- Mux pattern specification	
- ConduitFactory pattern specification	
- Adapter pattern specification	

Table 2: Activities of task "FVOPS specification & implementation"



The initial activities gather project-required information and its analysis. Knowledge about Conduit+ and JVOPS frameworks helps to extract hot spot of JVOPS and understand creation process of JVOPS-based network application. It is not possible create specialization patters without these knowledge because a specialization pattern definition is based on the analysis of the framework source code, UML diagrams and its documentation [6].

The Protocol Creator experiences are used in this stage, because the pilot version of FVOPS should demonstrate the same functionality as the Protocol Creator.

2.2.1 Results

At present time there was developed the pilot version of FVOPS where the integration of FRED and JVOPS has been done partly. It gives experiences to LUT research team in which direction implements the integration and it demonstrates possibility of the integration.

There was created specialization pattern in FRED notation that covers conduit Protocol. Together with the pattern, FRED guides and control application programmers in implementing communication protocol according to the convention of JVOPS. Programmers are able to automatically generate extension of the Protocol conduit; protocol states according to designed EFSM; set of information chunks; bind the protocol implementation with other conduits; and define actions on information chunk.

3 FVOPS evolution and related work

The research project FVOPS has not been finished even the first task is going on. The specialization patterns should be developed under conditions of *FVOPS specification & implementation*. Specialization patterns in FRED notation will cover following conduits: Mux, Adapter and ConduitFactory.

It is not enough to implement Protocol-Oriented Application Framework. Users of this framework should have imagination how it is used. The small but well-understandable application should demonstrate facilities of the framework. It is expected that authentication and service selection application based on Protocol-Oriented Application Framework will be developed under conditions of the third task (See Table 1).

3.1 Object-oriented evolution

An object-oriented technology has been developed recent years. It allows software to be rapidly developed, maintained, and reused. OO approaches widely used to design and



implement software products. FVOPS project should consider OO experience and give possibility to object-oriented designers use project results.

JVOPS hot spot should be depicted by UML notation. More important goal is present collaboration in UML between JVOPS classes that is significant in a stack of protocol development process. These results help to application developers understand process of protocol creation, reduce development time and resources.

FRED is a tool that guides and control application programmers in implementing applications according to the convention of a framework, but it uses own notation to specify framework hot spot. It follows that FRED users should use at least two design notations one for framework development and another for hot spot depiction. As was mentioned above, OO notation is became popular in recent years. It is necessary design mapping approach between FRED and UML notations.

3.2 JVOPS evolution

Conduit+ framework has specified four conduits: Protocol, Mux, Adapter and ConduitFactory. It is Conduit+ hot spot. They have been implemented in JVOPS but it also provides additional functionality that can be used by applications. These hot spot should be defined and depicted in FRED notation as well as the tool can guide and control application developer to use JVOPS functions.

The first JVOPS version was developed in 1997, and new versions with enhanced properties and documentation have been written in 1998, but since that time new version of Java has been developed. Sun Microsystems offers new technologies and APIs such as Micro Edition J2ME or Embedded Technology [7]. These Java changes should be considered in the direction that JVOPS could be used in communication protocol development for J2SE, J2ME, J2EE and embedded applications. Simulation and protocol testing possibility should be implemented in JVOPS as well as they required for solving protocol-oriented problems.



Reference

- [1] Hermann Huni, Ralph Johnson, Robert Engel, A Framework for Network Protocol Software, Association for Computing Machinery, Inc. 1995
- [2] Markku Hakala, Juha Hautamaki, Kai Koskimies, Jukka Paakki, Antti Viljamaa, Jukka Viljamaa, Generating Application Development Environments for Java Frameworks, Tampere University of Technology
- [3] <http://practise.cs.tut.fi/fred/>, 2001
- [4] Markku Hakala, Juha Hautamaki, Kai Koskimies, Jukka Paakki, Antti Viljamaa, Jukka Viljamaa, Annotating Reusable Software Architecture with Specialization Patterns, Tampere University of Technology
- [5] Pree W., Design Patterns for Object-Oriented Software Development, Addison-Wesley, 1995
- [6] Markku Hakala, Juha Hautamaki, Kai Koskimies, Jukka Paakki, Antti Viljamaa, Jukka Viljamaa, Architecture-Oriented Programming using FRED, Tampere University of Technology
- [7] <http://java.sun.com>, October 2001