A Study of Safety-Critical Java and its Specification Applied

Title:
A Study of Safety-Critical Java and its Specification Applied

Area:
Embedded and Distributed Systems

Field of Study:
Software Engineering

Project Period:
1^{st} September 2012 - 14^{th} January 2013

Project Group:
sw901e12

Participants:
Mikkel Todberg and Jeppe Lund Andersen

Supervisor(s):
René Rydhof Hansen, Andreas Dalsgaard

Number of Printed copies:
?

Number of Pages:
15

Java is one of the most known and widely used programming languages and several attempts have been made to bring Java into the world of Real-Time systems. Recently, Java has also been targeted the development of Safety-Critical systems with the new standard, JSR-302 Safety-Critical Java (SCJ), to facilitate the development of systems, certifiable with standards such as DO-178B Level A. The SCJ specification is still in draft, and lacks implementations and use cases in order to be thoroughly evaluated. This report presents a study of SCJ and the application of this for two use cases. We examine the draft version of the SCJ specification and an implementation of SCJ on the Java Optimized Processor (JOP). We take the specification and its implementation on JOP and apply this in the development of safety-critical software in collaboration with GomSpace that specialises in cubesat and nano-satellite platforms. We develop an SCJ version of their Cubesat Space Protocol (CSP), a network-layer delivery protocol for cubesats. We take this implementation and develop a Watchdog module in SCJ that utilises the CSP protocol to monitor other modules in a safety-critical environment and analyse the applicability of SCJ from a theoretical point of view with regard to theory in real-time software such as response-time analysis of applications. The result of this study is an analysis and findings on the applicability of SCJ in its current state, for the development of safety-critical systems with emphasis on transitioning ones Java experience to SCJ.

PREFACE

This project has been developed by five Software Engineering master students from Aalborg University, spring 2012. The report documents the development of a mobile application on the Android platform.

The report consists of five parts, excluding the appendix. The first part is an introductory part, which sets the boundaries and limitations of the project. Next there is a preliminary part in which the target mobile platform is investigated. The third part deals with the requirements and design of the application, before moving on to the fourth part which covers the development and implementation of the end product. Finally the project is concluded in a closing part, which discussed the perspectives and the suggestions on the developed application.

Source material referenced in this report will be notated with the initial letter of the surname of the author(s), followed by the year of publication. For example, [**?**], is a web page written by Egham published in 2011. The source refers to an entry in the bibliography list, where details regarding the source can be found.

Enjoy reading! Group sw804

CONTENTS

1 Introduction **5**

2 Real-Time Systems **6**

3 Safety-Critical Java **7**

4 Java Optimized Processor (JOP) **8**

5 The Cubesat Space Protocol (CSP) **9**

6 The CSP in Safety-Critical Java **10**

7 A CSP based Watchdog in Safety-Critical Java **11**

8 Reflection **12**

9 Future Works **13**

10 Conclusion **14**

CHAPTER

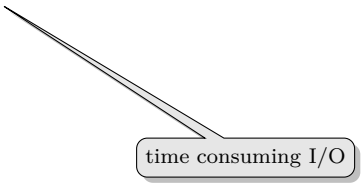
ONE

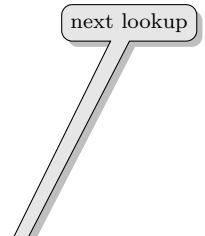
INTRODUCTION

Listing 1.1: A sample

```
#include <future>
std::map<std::string,std::string> french
{{"hello","bonjour"},{"world","tout le monde"}};
int main()
{
    std::string greet=french["hello"];
    auto f=std::async([&]{std::cout << greet <<", ";});
    std::string audience=french["word"];
    f.get();
    std::cout<<audience<<std::endl;
}

def main():
    hehe
```





CHAPTER

TWO

REAL-TIME SYSTEMS

CHAPTER

THREE

SAFETY-CRITICAL JAVA

CHAPTER

FOUR

JAVA OPTIMIZED PROCESSOR (JOP)

CHAPTER

FIVE

THE CUBESAT SPACE PROTOCOL (CSP)

CHAPTER

SIX

THE CSP IN SAFETY-CRITICAL JAVA

CHAPTER

SEVEN

A CSP BASED WATCHDOG IN SAFETY-CRITICAL JAVA

CHAPTER

EIGHT

REFLECTION

CHAPTER

NINE

FUTURE WORKS

CHAPTER

TEN

CONCLUSION

BIBLIOGRAPHY