UNIVERSITY OF SOFIA “St. KLIMENT OHRIDSKI”
FACULTY OF MATHEMATICS AND INFORMATICS

Abstract

University of Sofia “St.Kliment Ohridski”
Department of Information Technologies
Faculty of Mathematics and Informatics

Thesis title:
System for processing and extracting of meta data from a Java byte code.

Graduate student: Krasimir Ivanov Topchiyski, specialty “Computer Science”, Specialization “Information Systems”, faculty number M-21218

Supervisor: assoc.prof Boyan Bonchev, PhD, Sofia University, Department of Information Technologies, Faculty of Mathematics and Informatics

Defense of master’s thesis date: 14.02.2007

Key words: JDK, JVM, Java Bytecode, Annotations, Class Loading, Reflection API, Java EE 5, JEE Application, EJB, XML

Annotation:

The rapid development of the software systems and technologies, imposed Java as a powerful mean, that can be utilized for implementing almost every architectural software solution. The variety of solutions could span from mobile devices such as cellural phones, through elaborate corporate productive systems capable of processing millions of clients' requests worldwide. The growth of the Java technologies, the unceasing development of new standards in every software businesses, the increase of abstraction level of software products - all of these indicators are consequences of the evolutoin of Java as a programming language and also as reliable platform for developing state of the art software.

In the newest version of the Java language - Java Development Kit (JDK) 1.5, Java Enterprise Edition (Java EE) 5, the Java software vendor has provided an opportunity to embed meta info directly into the source code of the applications (these embedded code snippets are called "annotations"). This meta data provides the software developpers with means to link additional data to a class, field, method, method parameters or arbitrary piece of coding. In previous releases of the Java language, this data was stored in additional Extensible Markup Language (XML) files. The main concept behind this

thesis comes from the circumstance that it is not always possible to use the built in Java methods in order to access the meta data of an application (Reflection API). To achieve this, one should load all classes (Class Loading) in the Java Virtual Machine (JVM), which directly or indirectly are references in the class, which contains the meta data. This implies restrictions on the various scenarios where the meta data model should be used. For example, a class in an application cannot be loaded from the virtual machine, if it depends on another class, which cannot be loaded for some unexpected reason. This imposes the need of a system, which can extract the meta data from the Java Byte Code and structure it in convenient format, independently from the way in which data is loaded in the Java Virtual Machine.

The goal of this thesis is to implement a system, which provides mechanisms for operating on metadata depending on its object oriented representation.