Background

UML sequence diagrams display the message flow between the objects of an application for a particular and well defined interaction. Sequence diagrams are typically used as prescriptive models: they visualize the dynamic behaviour of a use case or are being used to evaluate static design models like class diagrams with regard to completeness of classes, methods, and associations.

In an iterative development process sequence diagrams become out-of-date very soon. Modifications in design, e.g. by enhancements or refactoring automatically lead to uselessness of the existing sequence diagrams. Especially in maintenance, the prescriptive sequence diagrams from the development are expired and obsolete.

The Reverse Engineering of sequence diagrams from existing software would be a significant advantage for development and maintenance. To analyse Java code, the tool “JDetex” for dynamic program analysis was developed at the Institute for Softwaretechnology at the University of Stuttgart. JDetex produces statement traces and static control flow information for Java systems. As output format of the Reverse Engineering, a standardized data format should be used to allow visualization of the sequence diagrams with existing UML tools.

Task description

Goal of this thesis is the Reverse Engineering of UML sequence diagrams from a given JDetex path trace and static control flow information. The sequence diagrams have to be stored as document-files which should be readable with existing UML Tools.

The available UML tools on the market have to be analysed in regard to their used data formats for model- and diagram storage and import capabilities for models and diagrams. Industrial standards, especially XMI, should be analysed how far they are capable of handling UML sequence diagrams. Based on this, an export format should be derived that is supported by common UML tools and corresponds to a specified and open standard.

Based on the given JDetex input a conversion to the specified output format should be implemented. In this implementation only UML 1.4 features for sequence diagrams should be used. The enhancements of UML 2.0 and their benefits for this approach have to be discussed. For the usage of the UML 2.0 enhancements the changes to the input format and the modifications to the output format should be specified.