# Automatically Recovering Design-Level UML Class Models from C++ Source Code

Kyle Swartz

Advisor: Prof. Jonathan I. Maletic

*Department of Computer Science, Kent State University, Kent Ohio 44242, U.S.A*

---

**Abstract**

*The work investigates methods for automatically recovering design-level Unified Modeling Language (UML) class diagrams from C++ source code. The ultimate goal is to support developers in better understanding the design of existing large scale software systems.*

*A set of mappings is proposed to identify various UML elements in terms of relationship types, multiplicities and aggregation semantics. The mappings used are based off of domain knowledge of the C++ language and common programming conventions and idioms. Additional analysis is conducted on these elements to detect design-level attributes of UML classes.*

*The result is a UML class model that reflects the actual design elements of an existing software system. This is critical for software developers because as software systems evolve and change over time the original design documentation becomes outdated and not reflective of the actual system. The output format of the class diagram is in yUML. This format is useful because it can be easily translated into a visual diagram. The srcML infrastructure supports the source code analysis and exploration needs. GraphVis is used to render the visual presentation of the resultant diagram. The results of this work will be compared to other reverse engineering tools.*

*Compared to other reverse engineering approaches this approach has the advantage of being very efficient and highly scalable. This is due in large part because of the underlying infrastructure, namely srcML. This allows for the approach to be seamlessly integrated into a developer's work environment to better support the construction and maintenance of large scale, high quality software.*

*Keywords:* Software engineering, reverse engineering, program comprehension, UML class models