SmartKT: A Search Framework to assist Program Comprehension using Smart Knowledge Transfer

Thesis submitted to the
Indian Institute of Technology Kharagpur
For award of the degree
of

Doctor of Philosophy

by

Srijoni Majumdar

Roll Number: 15AT92R01

Under the guidance of
Prof. Partha Pratim Das & Prof. Soumya Kanti Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India

Advanced Technology Development Centre
Indian Institute of Technology Kharagpur, India
September, 2022

ABSTRACT

Developers face significant program comprehension challenges while maintaining code due to the sparse availability of documentation, assistance tools, and help from the core development team. Approaches have been proposed to reverse engineer aspects related to control flow, low-level algorithmic details, design patterns, and the like. However, these do not provide a near-complete and multifaceted understanding of the design. The framework – SmartKT (Smart Knowledge Transfer) has been proposed to reverse engineer an integrated representation of design from C code repositories in the form of a knowledge graph and support semantic queries. To understand which design aspects should be known to fix bugs, or modify and add new code, studies have been conducted with developers from seven software companies that use C and work with different product areas. Through free-form interviews, directed examples, semi-structured questionnaire and case-studies, the perception of the developers have been captured in form of 90 representative design related queries. The aspects perceived relevant are mapping of application specific entities to code constructs, the behavior of application specific entities at runtime, dependencies among constructs, and the like, for which it is necessary to understand the structure, behavior, and the application specific / user interface (use case view) mapping of all code constructs. First, source code constructs are extracted with attributes like lexical and semantic scope, type, linkage, storage class using static instrumentation. These are further parsed and inspected to infer definition use chains, function references, type hierarchies, extern variable states, etc. Runtime traces are analyzed to infer memory access patterns, locality of reference, resource allocation and de-allocation, dynamic call graph, concurrency related aspects like thread activity, thread resource interaction patterns, concurrency induced issues like data race, live lock, deadlock, and the like. The required runtime events along with attributes are extracted using a dynamic instrumentation framework and further analyzed using decision rules or machine learning to infer higher order design elements. Natural language parsing has been used to mine application specific entities / operations from comments and source code constructs based on enumerated ontologies. Finally, for all code constructs, the information from source code, runtime events, and comment tokens are integrated syntactically using algorithms based on object code, symbol table, offsets, and semantically using ontologies, relations, and word vectors. Based on the knowledge graph, a query interface has been developed to support English language free-form template style semantic *design* queries. The design queries collected from the survey have been also used to test SmartKT over 7 open source Github projects. An average Precision of 84% and Recall of 82% have been achieved across multiple projects and queries, based on ground truth validated by industry experts. SmartKT addresses the challenges of *knowledge transfer* related to design through a simplistic code *design* search framework.

Keywords: Program Comprehension, Knowledge Graph, Code Instrumentation, Semantic Search, Recurrent Neural Networks, Sequence Models, Semantic Triples, Word Vectors