学 位 論 文 概 要

題　目　　Research on Construction and Applications of Formal Specification Component Attributes to Support Specification-Based Software Development

（仕様に基づくソフトウェア開発における形式仕様のコンポーネント属性の構築と応用に関する研究）

氏　名　　Jiandong Li

Most, if not all, of the phases in software development, such as program verification and validation, benefit from the use of formal specification. However, there still exist some problems in formal specification-based software development. The identified challenges are: (1) Components of a formal specification do have attributes and relationships hided in the specification, but they are not explicitly derived and described for making better sense of specification understanding and supporting subsequent development tasks. (2) How to prevent software faults in the phase of programming remains a challenge. Since software faults are costly to find and remove from programs, effective and proactive fault prevention approaches in coding are in high demand. (3) We are surprised to find that little work on automatically building trace links between formal specifications and code is done. Building effective trace links between formal specifications and their implementation is essential for conformance verification and program maintenance. (4) Formal specification-based code inspection (FSBCI) is a static technique for program verification. However, the program reading techniques used in the existing FSBCI methods, such as checklist-based reading, suffer from limited guidance and support for inspectors on fault detection. To increase the benefit of formal specification to software development and address these problems, this research takes the constituent components of formal specifications into account and constructs attributes for the specification components. On the basis of the designed attributes for the specification components, this research proposes: (1) a top-down approach to transform the formal specification into a novel machine-readable knowledge graph to provide comprehensible, well-organized details of the specification for stakeholder (e.g., programmer, inspector, tester) and computers. The transformation is done by extracting and storing information about attributes of each component and by establishing relationships between components in a formal specification. (2) a fault prevention approach in the programming phase of specification-based software development. It is characterized by the analysis of the dependences among components in a formal specification to derive an appropriate implementation order and the automatic generation of code fragments for the components in the specification using predefined transformation patterns. (3) an automated formal specification to code trace links establishment method. Its proposal is based on the common practice in specification-based implementation that the name and structure of a component and its relationships with others in the specification are often preserved in the implemented program, which are used to do similarity measurement. (4) a new code inspection method called Formal Specification Component Attributes-Based Code Inspection (FSCABCI). The essence of the proposed method is to use inspection to check whether each attribute of each component in the specification is correctly implemented by its corresponding component in code. Experiments are conducted to evaluate the proposed methods, respectively. The experiment results demonstrate that: (1) the proposed fault prevention approach can help the developers reduce the risk of introducing requirement-related errors and enhance their productivity. It is superior to Yu's method that develops a coding fault prevention guideline that describes examples of actual errors and the corrected code and trains the programmers before coding. (2) the proposed Formal-Specifications-to-Code trace links establishment method effectively builds the trace links between the components in specifications and the ones in code. It performs considerably better than two commonly used automatic methods, which are latent semantic indexing (LSI) and vector space model (VSM-cosine). (3) The proposed FSCABCI method can help inspectors to effectively detect requirements-related faults. It performs better than the existing formal specification-based inspection (FSBI) method, which checks whether functional scenarios in the specification are correctly implemented in the program. A supporting prototype tool is also developed to support the efficient use of the proposed approaches in practice.