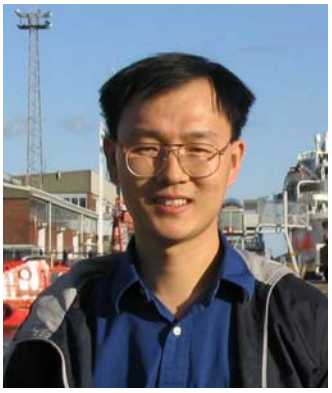

Fei Xie

Email: xie@cs.pdx.edu
Telephone: 503-725-2403

Open-Source

Research Interest

My research interests are primarily in the areas of software engineering and formal methods. I am particularly interested in development of formal method based techniques and tools for building safe, secure and reliable software systems. I am also interested in hardware/software co-design and co-verification.

Representative Research Projects

(All projects will lead to open-source tools supporting the proposed approaches.)

1. Security verification of component-based software systems

This project proposes, develops, and evaluates a novel approach to verification of security properties of component-based software systems. The unifying technology of this approach is automatic-translation based model checking enhanced by static analysis, compositional reasoning, and abstraction and supplemented by run-time monitoring. Model checking and component-based development (CBD) are strongly synergistic. Model checking leads to more secure components and systems composed from these components. CBD enormously reduces state spaces that model checkers must handle.

2. Enforcing non-functional properties of service-oriented grid-based computing

This project recognizes and exploits the synergism among several existing threads of research and integrates them into a potentially very powerful approach for solution of a new problem domain, assuring non-functional properties in the service-oriented grid architectures. It is a novel integration of theorem proving and model checking which extends them beyond their traditional realms of functional property validation and verification. It brings formal non-functional property assurance into grid computing.

3. Translation-based hardware and software co-verification

This project develops a translation-based approach to hardware and software co-verification of embedded systems. Software and hardware designs of an embedded system are translated into the input formal language of a state-of-the-art model checker and integrated via a bridge module that preserves the software and hardware semantics. Model checking is applied to the integrated formal model. Co-verification complexity is reduced through (1) leveraging reduction algorithms of the model checkers, (2) applying reduction algorithms in translation as transformations, and (3) conducting compositional reasoning across interfaces of the bridge module. Our approach has been implemented for co-verification of software designs in Executable UML and hardware designs in Verilog.