Predictive access control for distributed computation
We show how to use aspect-oriented programming to separate security and trust issues from the logical design of mobile, distributed systems. The main challenge is how to enforce various types of security policies, in particular predictive access control policies — policies based on the future behavior of a program. A novel feature of our approach is that we can define policies concerning secondary use of data.
Combining Static Analysis and Runtime Checking in Security Aspects for Distributed Tuple Spaces

Enforcing security policies to distributed systems is difficult, in particular, to a system containing untrusted components. We designed AspectKE*, an aspect-oriented programming language based on distributed tuple spaces to tackle this issue. One of the key features in AspectKE* is the program analysis predicates and functions that provide information on future behavior of a program. With a dual value evaluation mechanism that handles results of static analysis and runtime values at the same time, those functions and predicates enable the users to specify security policies in a uniform manner. Our two-staged implementation strategy gathers fundamental static analysis information at load-time, so as to avoid performing all analysis at runtime. We built a compiler for AspectKE*, and successfully implemented security aspects for a distributed chat system and an electronic healthcare record workflow system.

General information
State: Published
Organisations: Language-Based Technology, Department of Informatics and Mathematical Modeling, Japan Advanced Institute of Science and Technology, University of Tokyo
Authors: Yang, F. (Intern), Aotani, T. (Ekstern), Masuhara, H. (Ekstern), Nielson, F. (Intern), Nielson, H. R. (Intern)
Pages: 202-218
Publication date: 2011
AspectKE*: Security Aspects with Program Analysis for Distributed Systems

Enforcing security policies to distributed systems is difficult, in particular, when a system contains untrusted components. We designed AspectKE*, a distributed AOP language based on a tuple space, to tackle this issue. In AspectKE*, aspects can enforce access control policies that depend on future behavior of running processes. One of the key language features is the predicates and functions that extract results of static program analysis, which are useful for defining security aspects that have to know about future behavior of a program. AspectKE* also provides a novel variable binding mechanism for pointcuts, so that pointcuts can uniformly specify join points based on both static and dynamic information about the program. Our implementation strategy performs fundamental static analysis at load-time, so as to retain runtime overheads minimal. We implemented a compiler for AspectKE*, and demonstrate usefulness of AspectKE* through a security aspect for a distributed chat system.

General information
State: Published
Organisations: Language-Based Technology, Department of Informatics and Mathematical Modeling, University of Tokyo
Authors: Yang, F. (Intern), Masuhara, H. (Ekstern), Aotani, T. (Ekstern), Nielson, F. (Intern), Nielson, H. R. (Intern)
Number of pages: 47
Pages: 27-31
Publication date: 2010

Host publication information
Title of host publication: Proceedings of the 9th Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS '10)
Publisher: Universitätsverlag Potsdam
Editors: Adams, B., Haupt, M., Lohmann, D.
ISBN (Print): 978-3-86956-043-4
Series: Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam
Number: 33
Main Research Area: Technical/natural sciences
Conference: 9th Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS '10), Rennes, France, 01/01/2010
Aspect Oriented Programming, Program Analysis, Security Policies, Distributed Systems, Tuple Spaces
Electronic versions:
tbhpi33_S27_31.pdf
Links:
Source: orbit
Source-ID: 259697
Publication: Research - peer-review › Article in proceedings – Annual report year: 2010

AspectKE*: Security Aspects with Program Analysis for Distributed Systems

AspectKE* is the first distributed AOP language based on a tuple space system. It is designed to enforce security policies to applications containing untrusted processes. One of the key features is the high-level predicates that extract results of static program analysis. These predicates provide users an easy way to define aspects by providing information about future behavior of processes, which are shown to be useful to implement security policies such as secrecy and integrity. The users of AspectKE* do not need to write low-level analysis by themselves. In the demonstration, we show basic features of AspectKE* and a case study of building a secure distributed chat application that contains a malicious process.

General information
State: Published
Organisations: Language-Based Technology, Department of Informatics and Mathematical Modeling, University of Tokyo
Authors: Yang, F. (Intern), Masuhara, H. (Ekstern), Aotani, T. (Ekstern), Nielson, F. (Intern), Nielson, H. R. (Intern)
Publication date: 2010

Publication information
Original language: English
Place of publication: Demonstration track of the 9th International Conference on Aspect-Oriented Software Development (AOSD'10), Rennes and Saint Malo, France
Main Research Area: Technical/natural sciences
Aspect Oriented Programming, Program Analysis, Distributed Systems, Security Policies, Tuple Spaces
Aspects with Program Analysis for Security Policies

Enforcing security policies to IT systems, especially for a mobile distributed system, is challenging. As society becomes more IT-savvy, our expectations about security and privacy evolve. This is usually followed by changes in regulation in the form of standards and legislation. In many cases, small modification of the security requirement might lead to substantial changes in a number of modules within a large mobile distributed system. Indeed, security is a crosscutting concern which can spread to many business modules within a system, and is difficult to be integrated in a modular way. This dissertation explores the principles of adding challenging security policies to existing systems with great flexibility and modularity. The policies concerned cover both classical access control and explicit information flow policies. We built our solution by combining aspect-oriented programming techniques with static program analysis techniques. The former technique can separate security concerns out of the main logic, and thus improves system modularity. The latter can analyze the system behavior, and thus helps detect software bugs or potential malicious code. We present AspectKE, an aspect-oriented extension of the process calculus KLAIM that excels at modeling mobile, distributed systems. A novel feature of our approach is that advices are able to analyze the future use of data, which is achieved by using program analysis techniques. We also present AspectK to propose other possible aspect-oriented extensions based on KLAIM, followed by a discussion of open joinpoints that commonly exist in coordination languages such as KLAIM. Based on the idea of AspectKE, we design and implement a proof-of-concept programming language AspectKE*, which enables programmers to easily specify analysis-based security policies with the help of high-level program analysis predicates and functions. The prototype is efficiently realized by a two-stage implementation strategy and a static-dynamic dual value evaluation mechanism. We have performed two case studies to evaluate our programming model and language design. One application is based on an electronic health care workflow system. The other is a distributed chat system. We considered a number of security policies for both primary and secondary use of data, classical access control and predictive access control - control access based on the future behavior of a program. Some of the above mentioned policies can only be enforced by analysis of process continuations.

General information
State: Published
Organisations: Language-Based Technology, Department of Informatics and Mathematical Modeling
Authors: Yang, F. (Intern), Nielson, F. (Intern), Nielson, H. R. (Intern)
Publication date: 2010

Publication Information
Place of publication: Kgs. Lyngby, Denmark
Publisher: Technical University of Denmark (DTU)
Original language: English
Series: IMM-PHD-2010-239
Main Research Area: Technical/natural sciences
Electronic versions: phd239_fy.pdf
Source: orbit
Source-ID: 266345
Publication: Research › Ph.D. thesis – Annual report year: 2010

Advice for Coordination

We show how to extend a coordination language with support for aspect oriented programming. The main challenge is how to properly deal with the trapping of actions before the actual data have been bound to the formal parameters. This necessitates dealing with open joinpoints – which is more demanding than the closed joinpoints in more traditional aspect oriented languages like AspectJ. The usefulness of our approach is demonstrated by mechanisms for discretionary and mandatory access control policies, as usually expressed by reference monitors, as well as mechanisms for logging actions.

General information
State: Published
Organisations: Language-Based Technology, Department of Informatics and Mathematical Modeling, Imperial College London
Authors: Hankin, C. (Ekstern), Nielson, F. (Intern), Nielson, H. R. (Intern), Yang, F. (Intern)
Number of pages: 346
Pages: 153-168
Publication date: 2008

Host publication information
Analyzing the Control Structure of PEPA

The Performance Evaluation Process Algebra, PEPA, is introduced by Jane Hillston as a stochastic process algebra for modelling distributed systems and especially suitable for performance evaluation. We present a static analysis that very precisely approximates the control structure of processes expressed in PEPA. The analysis technique we adopted is Data Flow Analysis. We begin the analysis by defining an appropriate transfer function, then with the classical worklist algorithm we construct a finite automaton that captures all possible interactions among processes. By annotating labels and layers to PEPA programs, the approximating result is very precise. Based on the analysis, we also develop algorithms for validating the deadlock property of PEPA programs. The techniques have been implemented in a tool which is able to analyze processes with a control structure that more than one thousand states.

General information

State: Published
Organisations: Department of Informatics and Mathematical Modeling, Computer Science and Engineering, Language-Based Technology
Authors: Yang, F. (Intern), Nielson, H. R. (Intern)
Event: Poster session presented at The 14th International Static Analysis Symposium (SAS 2007), Lyngby, Denmark.
Main Research Area: Technical/natural sciences
PEPA, Date Flow Analysis, Control Structure, Finite Automaton, Deadlock, Static Analysis, Stochastic Process Algebra
Source: orbit
Source-ID: 259698
Publication: Research › Poster – Annual report year: 2007

Projects:

Aspects for security policies

Department of Informatics and Mathematical Modeling
Period: 01/06/2007 → 08/12/2010
Number of participants: 6
Phd Student:
Yang, Fan (Intern)
Supervisor:
Nielson, Hanne Riis (Intern)
Main Supervisor:
Nielson, Flemming (Intern)
Examiner:
Probst, Christian W. (Intern)
De Nicola, Rocco (Ekstern)
Südholt, Mario (Ekstern)

Financing sources

Source: Internal funding (public)
Name of research programme: DTU-lønnet stipendie
Project: PhD