Specification-time Conflict Detection and Resolution in Usage Control Policies

Master thesis

Supervisors: Prof. Dr. Alexander Pretschner, Prachi Kumari
Email: {prachi.kumari} @ in.tum.de
Phone: +49 89 289 – 17378
Starting date: immediately

Context and Goal
Usage control is a generalization of access control that specifies and enforces what must or must not happen to data once access to it has been granted. Some example policies include "delete data after 30 days", "don't distribute my data", "notify me whenever somebody accesses the data", "don't make more than 5 copies". These policies must be enforced at and across each layer of abstraction in each machine in a distributed system. Conflicts can arise in the set of policies. The purpose of this thesis is to look into all conflicts that can be detected statically (that is, at the policy specification time), and come up with an automated detection and resolution of them.

Description
Conflicts in usage control policies can arise from various sources. For example, one policy might define a duty that includes performing certain actions that are forbidden by another policy, as in "no non-anonymised data should leave the system" vs. "report complete logs (including user profile) whenever this data is accessed". Conflicts can also arise when policies put constraints upon an action that targets an object which is a collection of other objects or forms part of another object in the domain data model. For example, an album consists of several photos where each photo might be associated with a different policy ("copy max 5 times", "copy upon payment", "copy but notify", "copy with lower quality" etc.). In this case, a policy for the album that constraints copy action might be in conflict with one or more of the policies for photos. Apart from object hierarchy, policy conflicts can also arise from subjects' hierarchy: for example, rights and duties of subjects at one level can consist of rights and duties of subjects at other levels; and because of several other reasons. Intuitively, detecting these conflicts at the time of policy specification is not very difficult. Although, in a large enterprise, it might be challenging for an administrator to manually compare a new policy with all the existing ones and detect and resolve any conflicts within realistic time limits. For this reason, a tool support for the process is required. The more interesting part of this work would be to see how some patterns could be formed for the conflicts; if for all such cases, a hierarchical order could be assigned to the policies (or to classes of policies) and if these patterns and hierarchies (and other conflict resolution strategies) could be used to successfully resolve policy conflicts automatically.

Expected results and Evaluation
In this thesis, the student will start by getting familiar with usage control concepts. Then he/she should study related work on policy conflicts and their static resolution, look into the specification–level policies and their classes for usage control and develop a taxonomy of conflicts for usage control policies. He/she will then come up with several conflict detection and resolution strategies based on which, the outcome of this work will be a tool for automated conflict recognition and resolution. The work will be evaluated with a case study.
Workplan

1. Familiarize yourself with the general usage control concepts [1 - 3].
2. Familiarize yourself with the existing work on policy conflicts. A good starting point would be [4].
3. Come up with a policy conflict resolution scheme for usage control.
4. Develop tool support for automatic conflict detection and resolution.
5. Evaluate the results/tools with a case study.
6. Write a dissertation and submit it along with raw code (and executables, if applicable), virtual machines and proper supporting documentation.

References


