Security Policies in Constraint Handling Rules

Position Paper

Thom Frühwirth
Ulm University, Germany
http://www.informatik.uni-ulm.de/pm/fileadmin/pm/home/fruehwirth/

Abstract

Role-Based Access Control (RBAC) is a popular security policy framework. User access to resources is controlled by roles and privileges. Constraint Handling Rules (CHR) is a rule-based constraint logic language. CHR has been used to implement security policies for trust management systems like RBAC for more than two decades.

Constraint Handling Rules (CHR) [Frü09, Frü18, Frü15, FR18] is a logical rule-based language and framework employing constraints. In this short paper, we describe work on RBAC that is implemented in CHR. We just cite the main works for each group of authors. Further references may be found in the cited papers and/or by googling.

Based on [BFM02], Bistarelli et. al. [BMS10, BMS12] apply an extension of Datalog by weighted facts to model role-based trust management. Deduction can validate access requests. Abduction can compute missing credentials if the access is denied and it can compute the level of preference that would grant the access. Both deduction and abduction are expressed in Weighted Datalog and translated into CHR for execution. [BCMS14] show how this deductive and abductive reasoning can be efficiently ported to Android enabling distributed authorization. Both deduction and abduction are implemented as programs in a version of CHR that is embedded into Java (JCHR).

Ribeiro et. al. [RG99] present a static analyzer that automatically verifies consistency of workflow specifications written in WPDL (Workflow Process Definition Language) and of specifications in a security policy language (SPL). The analyzer is implemented with CHR embedded in SICstus Prolog. [RZFG00] further describes this Policy Consistency Verifier (PVC). It now includes constraints automatically annotated with temporal information. [RF07] presents further work on the security policy language (SPL). It can express the concepts of permission and prohibition, and some restricted forms of obligation as well as history-based approaches. Given a SPL specification, it is verified using CHR and then compiled to Java into a corresponding security access monitor. The current CHR verifier has about 300 rules and is able to solve all SPL constraints, including the constraints implicitly qualified with time.

The Object Constraint Language (OCL) is a declarative text language describing rules applying to Unified Modeling Language (UML) models. OCL provides constraint and object query expressions on models that cannot be expressed by diagrammatic notation. OCL is now a key component of the new OMG standard recommendation for transforming models. Model finders automatically verify UML/OCL models by checking satisfiability (consistency) of models using example instances. The work of [DTVH16] presents oclCHR `https://uet.vnu.edu.vn/~hanhdd/oclCHR/`, a verifier implemented in CHR embedded in Eclipse Prolog. It is of interest here, because the authors use an UML model of RBAC as their main example.

Finally, we would like to cite two approaches of RBAC in logical languages that can be readily translated into CHR. [BS03] show how a range of role-based access control (RBAC) models may be usefully represented as executable logical specifications in constraint logic programs (CLP). Like Weighted Datalog, CLP clauses can be translated to CHR [Frü09].

[OPR18] presents a declarative interpretation of Access Permissions (APs) as Linear Concurrent Constraint Programs (LCC). By interpreting LCC programs as formulae in intuitionistic linear logic, they can verify properties of AP programs such as deadlock detection, correctness, and concurrency. CHR also admits a linear logic interpretation [Bet14] and is closely related to the more recent LCC language. Translations between LCC and CHR are given in [Mar10].

Concluding, CHR is a often used language to build reasoning services. In this paper, we showed this surveying shortly work on the problem of security policies, i.e. access control. We would like to thank the anonymous referees for their helpful comments.

References

[BCMS14] Stefano Bistarelli, Gianpiero Costantino, Fabio Martinelli, and Francesco Santini. An improved role-based access to android applications with jchr. In *Availability, Reliability and Security (ARES), 2014 Ninth International Conference on*, pages 341–348. IEEE, 2014.

[Bet14] Hariolf Betz. *A unified analytical foundation for constraint handling rules*. BoD, 2014.

[BFM02] Stefano Bistarelli, Thom Frühwirth, and Michael Marte. Soft constraint propagation and solving in chrs. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 1–5. ACM, 2002.

[BMS10] Stefano Bistarelli, Fabio Martinelli, and Francesco Santini. A formal framework for trust policy negotiation in autonomic systems: Abduction with soft constraints. In *International Conference on Autonomic and Trusted Computing*, pages 268–282. Springer, 2010.

[BMS12] Stefano Bistarelli, Fabio Martinelli, and Francesco Santini. A semiring-based framework for the deduction/abduction reasoning in access control with weighted credentials. *Computers & Mathematics with Applications*, 64(4):447–462, 2012.

[BS03] Steve Barker and Peter J Stuckey. Flexible access control policy specification with constraint logic programming. *ACM Transactions on Information and System Security (TISSEC)*, 6(4):501–546, 2003.

[DTVH16] Duc-Hanh Dang, Anh-Hoang Truong, and Dang Van Hung. On model finding with constraint patterns. In *SoMeT*, pages 279–290, 2016.

[FR18] Thom Frühwirth and Frank Raiser. *Constraint Handling Rules-Compilation, Execution, and Analysis: Large Print Edition.* BoD, 2018.

[Frü09] Thom Frühwirth. *Constraint handling rules.* Cambridge University Press, 2009.

[Frü15] Thom Frühwirth. Constraint handling rules – what else? In *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, pages 13–34. Springer, 2015.

[Frü18] Thom Frühwirth. *The CHR Web Site – `www.constraint-handling-rules.org`*. Ulm University, 2018.

[Mar10] Thierry Martinez. Semantics-preserving translations between linear concurrent constraint programming and constraint handling rules. In *Proceedings of the 12th international ACM SIGPLAN symposium on Principles and practice of declarative programming*, pages 57–66. ACM, 2010.

[OPR18] Carlos Olarte, Elaine Pimentel, and Camilo Rueda. A concurrent constraint programming interpretation of access permissions. *Theory and Practice of Logic Programming*, pages 1–44, 2018.

[RF07] Carlos Ribeiro and Paulo Ferreira. A policy-oriented language for expressing security specifications. *IJ Network Security*, 5(3):299–316, 2007.

[RG99] Carlos Ribeiro and Paulo Guedes. Verifying workflow processes against organization security policies. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 1999.(WET ICE'99) Proceedings. IEEE 8th International Workshops on*, pages 190–191. IEEE, 1999.

[RZFG00] Carlos Ribeiro, André Zúquete, Paulo Ferreira, and Paulo Guedes. Security policy consistency. In T. Fruehwirth et al., editors, *RCoRP '00: Proc. 1st Workshop on Rule-Based Constraint Reasoning and Programming*, 2000. arXiv preprint cs/0006045.