Modeling and Typing Authorizations in Role-Based Interaction

Silvia Ghilezan[1] Svetlana Jakšić[1] Jovanka Pantović[1] Jorge A. Pérez[2]
Hugo T. Vieira[3]
[1]University of Novi Sad, Serbia
[2]University of Groningen, The Netherlands
[3]IMT School for Advanced Studies Lucca, Italy

In this document we summarize the achievements reported in a collection of papers [3, 4, 5], work carried out in the context of COST Action IC12012 BETTY, and supported by in particular via STSM-IC1201-070713-32410, STSM-IC1201-090214-038830, STSM-IC1201-180115-054522, and STSM-IC1201-240416-072568.

Abstract

Communication intensive systems are nowadays part of lives as never before and therefore ensuring secure correct interactions is of crucial importance. Interactions between communicating parties are often specified in terms of protocols that identify the roles engaged in the communications. This may be so for security purposes, as different roles may have different communication permissions, or for behavioral purposes, as the overall system behavior may be captured in terms of the individual contributions of the roles involved. However, it is also interesting to notice that sometimes the behavioral roles may actually be carried out by distributed parties, for instance, when a server delegates a subtask to a slave transparently to the client: the *role* of the server and of the slave is the same from the point of view of the client. When reasoning about secure interactions, taking into account a party should be *authorized* to perform an action on behalf of a role, such dynamic role impersonation is a concern. On the one hand, it is not reasonable to expect a party to share its authentication credentials with anyone else. On the other hand, there must somehow be a mechanism that allows authorizations to be shared among the parties impersonating a role.

In [3] we introduce a model that allows to reason about authorization sharing via communication, together with an extension of an existing behavioral type system [1] in order to statically ensure that, at runtime, communications will always be carried out by properly authorized parties. We build on the π-calculus and add to the communication actions labels, that identify the role in which processes are to interact, and a tag to denote if the action is authorized or not. We also add primitives that allow to communicate authorizations so that processes can acquire authorizations that are not initially held by them. The behavioral type system is enriched so as to capture the role authorization communication which, in particular, allows to reason about such communications at the level of the type specification.

Based on the preliminary ideas developed in [3] we also consider the problem of disciplining authorized interaction, considering role specifications are absent, in the presence of authorization *delegation* [4], i.e., when communicating an authorization actually causes the emitting party to lose the authorization. This mechanism seems quite natural in the setting of *sessions* where delegation has been around since the initial proposal [6]. We introduce an algebraic characterization of such mechanism, namely by introducing a language construct to represent an *authorization scope*, defining the operational semantics (both via a labelled-transition system and via reduction together with structural congruence) and ensuring that a standard notion of observational equivalence is a congruence w.r.t. all language constructs, as well as considering a preliminary exploration of some behavioral axioms. We believe that scope authorization is an interesting novel construct that is worthwhile exploring, a conviction reinforced by the particular technical characterization of the language construct (which evidenced its distinction w.r.t. other known scoping operators). We also develop a static analysis technique that ensures system interactions are always authorized, even in the presence of delegation.

In [5] we provide a unified presentation of the approaches reported in [3, 4]. While the paper [3] already considers dynamic role authorization in the setting of conversation types, the paper [4] introduces the idea of authorization scopes to control and give a spatial meaning to explicit authorizations on process actions, which inspires the work presented in [5]. As such, the papers [3] and [4] offer orthogonal perspectives to the issue of dynamic role authorization with delegation; [5] combines these two perspectives in a uniform way, adding a labeled transition system semantics supporting authorization (not present in the previous papers), together with a preliminary investigation of the associated behavioral theory, and we extend the type system (based on conversation types [2]) in order to account for explicit role authorization and authorization scopes.

References

[1] P. Baltazar, L. Caires, V. T. Vasconcelos, and H. T. Vieira. A type system for flexible role assignment in multiparty communicating systems. In C. Palamidessi and M. D. Ryan, editors, *Trustworthy Global Computing - 7th International Symposium, TGC 2012, Revised Selected Papers*, volume 8191 of *LNCS*, pages 82–96. Springer, 2012.

[2] L. Caires and H. T. Vieira. Conversation types. *Theor. Comput. Sci.*, 411(51-52):4399–4440, 2010.

[3] S. Ghilezan, S. Jaksic, J. Pantovic, J. A. Pérez, and H. T. Vieira. Dynamic role authorization in multiparty conversations. In M. Carbone, editor, *Proceedings Third Workshop on Behavioural Types, BEAT 2014*, volume 162 of *EPTCS*, pages 1–8, 2014.

[4] S. Ghilezan, S. Jaksic, J. Pantovic, J. A. Pérez, and H. T. Vieira. A typed model for dynamic authorizations. In S. Gay and J. Alglave, editors, *Proceedings Eighth International Workshop on Programming Language Approaches to Concurrency- and Communication-cEntric Software, PLACES 2015*, volume 203 of *EPTCS*, pages 73–84, 2015.

[5] S. Ghilezan, S. Jaksic, J. Pantovic, J. A. Pérez, and H. T. Vieira. Dynamic role authorization in multiparty conversations. *Formal Asp. Comput.*, 28(4):643–667, 2016.

[6] K. Honda, V. T. Vasconcelos, and M. Kubo. Language primitives and type discipline for structured communication-based programming. In C. Hankin, editor, *Programming Languages and Systems - ESOP'98, 7th European Symposium on Programming, Proceedings*, volume 1381 of *LNCS*, pages 122–138. Springer, 1998.