MATYSS: Models And TYpes for Secure Sessions

COLOR'09 proposal

November 13, 2008

1 Object of the proposal

The goal of the project is to study models and type systems for safe and secure sessions. A *session* is an abstraction for various forms of "structured communication" which may occur in a parallel and distributed computing environment. Examples of sessions are a client-service negotiation, a financial transaction, or a multiparty interaction among different services within a web application.

Language-based support for sessions has now become the subject of active research. Primitives for enabling programmers to code sessions in a flexible way, as well as type systems (*session types*) ensuring the compliance of programs to session specifications, have already been studied for various calculi and languages.

The key properties ensured by session types are the *consistency* of the communication patterns exhibited by the various partners (implying the absence of communication errors at run time), and *progress*, assuring the absence of deadlock, so that each terminating session completes with success.

On the other hand, little attention has been paid so far to security issues within session types. This is an evident weakness, since properties such as confidentiality and integrity of data become all the more crucial in an open, unreliable environment such as that of global networks, where communicating partners may not trust each other.

The MATYSS project will address the question of incorporating secure information flow requirements within session types, in the setting of a name-passing calculus (akin to the π-calculus) with asynchronous communication and multiparty sessions.

2 Participating teams

- **MIMOSA team**, INRIA Sophia Antipolis Méditerranée.

  Participants: Ilaria Castellani (chargée de recherche INRIA, action leader) and Tamara Rezk (chargée de recherche INRIA). Web pages:

http://www-sop.inria.fr/mimosa/Ilaria.Castellani/

http://www-sop.inria.fr/everest/Tamara.Rezk/

- **"Semantics and Logic of Computation" group** (more specifically, the subgroup on "Types for higher-order, concurrent, and object-oriented processes"), Computer Science Department, University of Torino, Corso Svizzera 185, I-10149 Torino, Italy.

  Participants: Mariangiola Dezani-Ciancaglini (full professor, site leader) and Sara Capecchi (post-doctoral fellow). Web pages:

  http://www.di.unito.it/~dezani/index.html

  http://www.di.unito.it/~capecchi/

3 Scientific description

With the advent of the web and multicore architectures, and the proliferation of programmable and interconnectable devices in our living and working environments, we are faced today with a powerful and heterogeneous computing environment. This environment is inherently parallel and distributed and, unlike previous programming scenarios, it heavily relies on communication. It calls for a new programming paradigm which is sometimes qualified as *communication-centred* or *communication-intensive*. Moreover, since computations take place concurrently on all kinds of different devices, controlled by parties which possibly do not trust each other, and communication occurs in an open, unreliable global space, security properties such as confidentiality and integrity of data become of crucial importance. The issue is then to develop models, as well as programming abstractions and methodologies, to be able to exploit the rich potential of this global computing environment, while at the same time making sure that we can harness its complexity and get around its many "security holes". To this end, models and languages for web applications and global computing will have to be *security-minded* from their very conception, and make use of types for communication and data structures (such as session types), as well as for security.

A session abstracts a sequence of heterogeneous messages exchanged between various participants. These messages can be interleaved with computations executed by the single participants. A case in point is delegation of activities to third parties, which often occurs transparently to the participants. The session abstraction should enable the programmer to code complex communication structures in a flexible way. To be of full use, it should come equipped with some specification or type (session type), providing the programmer with a way of checking the desired properties of sessions before engaging into them.

Sessions and session types have been developed for various programming paradigms, including name-passing calculi such as the pi-calculus [THK94, HVK98, YV07], where session types were originally introduced, multithreaded functional languages [VRG04, VGR06], mobile ambients [GCDC06], object-oriented languages [DCMYD06, DCDGY07, CCDC+08, BCDC+08] and web description languages [CHY, CHY07]. Type systems guaranteeing that values of the expected type are exchanged in the required order, as well as the absence of deadlocks, have been proposed for both binary [CDCY07, DCdLY08]

and multiparty sessions [HYC08, BCD+08], in both synchronous and asynchronous communication scenarios.

The question of ensuring *secure information flow* in a system with multiple security levels, first raised in the early eighties [GM82], regained a great deal of interest in the last decade, due to the evolution of the computing environment. The question has now been studied in some depth both for programming languages (see [SM03] for a review) and for process calculi [RS99, FG01, HVY00, Pot02, HR02, HY07, Hen04, BCR04, Kob05, CR05, Cas07]. In a programming language, assuming two levels of confidentiality (secret and public), this property amounts to requiring that public outputs of programs should not be influenced by their secret inputs. In process calculi, it roughly prescribes that public actions of processes should not depend on secret actions. However, what is to be taken as "dependency" inside sessions may be quite subtle, since channels may be used in different ways (e.g. in a linear way) and the flow of data and the flow of control are closely intertwined in process calculi.

The aim of the MATYSS project is to develop type systems for assuring confidentiality and integrity of data in multiparty asynchronous sessions with delegation, taking into account the proposal of [Kol08], where a simplified form of binary and synchronous sessions without delegation is considered. This research will naturally build on the previous work carried out by the Torino team on session types for various core languages and process calculi [DCMYD06, GCDC06, DCDGY07, CDCY07, CCDC+08, DCdLY08, BCDC+08, BCD+08], and by the MIMOSA team on types for secure information flow in concurrent languages and calculi [BC02, ABC07, AB05, BK07, Cas07].

3.1 Expected impact

The emergence of web programming and multicore processors is inducing a fundamental shift in the software development paradigm, by placing communication and concurrency at the centre of the computing activity. This new paradigm, however, still lacks a mature programming methodology. If sequential programming may be hard, especially when coding critical applications where safety and security are essential requirements, concurrent communication-centred programming in an open network is harder, because it exposes programmers and designers to the complexity arising from the composition of communication behaviours which include the possibility of deadlock, livelock and different forms of partial failure, and which are liable to security attacks.

The development of effective and secure programming methodologies for the communication centred computing paradigm demands exploration and understanding of a wide variety of ideas and techniques, among which models and types for safe and secure sessions will play an important role.

4 Modalities of the collaboration

We plan to hold 4 plenary meetings, alternately in Sophia Antipolis and in Torino. Longer visits of each of the participants to the other team are also programmed, with an expected duration of one week to one month (cf Section 5 for more details).

Specificities and complementarity of the teams

- The MIMOSA team has a long-standing experience in models for concurrent computation. Since year 2001, Ilaria Castellani has been working on type systems for ensuring security properties in concurrent languages and process calculi. Tamara Rezk, a former member of the EVEREST team who recently joined MIMOSA (after the closure of EVEREST), has been working since 2004 on language-based security and on compilers.

- The research team on "Types for higher-order, concurrent, and object-oriented processes" has given important contributions in the field of type systems and static analysis tools for calculi and core languages. From 2005 Mariangiola Dezani (leader of this team) has worked mainly on session types for assuring safety of communication protocols. Sara Capecchi joined this team in 2007: her previous research activity was focused on safe extensions of class-based languages for gaining flexibility. Recently she collaborated with Mariangiola Dezani on session types for object-oriented languages.

- The MIMOSA and Torino teams have not been previously working together on the subject of this proposal. In the past they have both participated to the EU FET Global Computing project MIKADO. Gérard Boudol (Research Director within the MIMOSA team) and Mariangiola Dezani are the co-promoters of the PhD studies of Marija Kolundžija. The subject of the PhD thesis of Marija Kolundžija is type systems for access control and information flow. The defence is expected to take place in February 2009.

5 Requested resources

A total amount of approximately **11 K Euros** is requested for the project. This funding will be used to cover travel and accomodation expenses for the members of the project, when attending project's meetings and during the longer visits paid by single members to the other team. More precisely, the expected expenses are:

- 4 plenary meetings, alternately in Sophia Antipolis and Torino: **3200 Euros**.
- 1 half-month visit by Mariangiola Dezani to Sophia Antipolis: **1800 Euros**.
- 1 one-month visit by Sara Capecchi to Sophia Antipolis: **3600 Euros**.
- 1 one-week visit by Ilaria Castellani to Torino: **1190 Euros**.
- 1 one-week visit by Tamara Rezk to Torino: **1190 Euros**.

6 Interaction with other projects

Some interaction is expected with the following French projects:

1. ANR project PARSEC (including both Ilaria Castellani and Tamara Rezk), started on January 1st, 2007:
   http://moscova.inria.fr/~zappa/projects/parsec/meetings.html

2. ANR project PARTOUT (including Ilaria Castellani), starting on January 1st, 2009.

7 Previous COLOR fundings

The MIMOSA team has not benefited from any previous COLOR funding.

References

[ABC07] A. Almeida Matos, G. Boudol and I. Castellani. Typing Noninterference for Reactive Programs. *Journal of Logic and Algebraic Programming* 72: 124-156, 2007.

[AB05] A. Almeida Matos and G. Boudol. On Declassification and the Non-disclosure Policy. In *CSFW'05*, 226–240. IEEE Computer Society, 2005.

[BC02] G. Boudol and I. Castellani. Noninterference for Concurrent Programs and Thread Systems. *Theoretical Computer Science* 281(1): 109-130, 2002.

[BCD+08] L. Bettini, M. Coppo, L. D'Antoni, M. De Luca, M. Dezani-Ciancaglini, and N. Yoshida. Global Progress in Dynamically Interleaved Multiparty Sessions. In *CONCUR'08*, *LNCS* 5201: 418–433. Springer-Verlag, 2008.

[BCDC+08] L. Bettini, S. Capecchi, M. Dezani-Ciancaglini, E. Giachino, and B. Venneri. Session and Union Types for Object Oriented Programming. In *Concurrency, Graphs and Models*, *LNCS* 5065: 659–680. Springer-Verlag, 2008.

[BCR04] A. Bossi, R. Focardi, C. Piazza and S. Rossi. Verifying Persistent Security Properties. *Computer Languages, Systems and Structures* 30(3-4): 231-258, 2004.

[BK07] G. Boudol, M. Kolundzija. Access Control and Declassification. In *CNS'07*, *CCIS* 1: 85–98. Springer-Verlag, 2007.

[Cas07] I. Castellani. State-oriented Non-interference for CCS. In *SecCo'07*, *ENTCS*, 194(1): 39–60. Elsevier, 2007.

[CCDC+08] S. Capecchi, M. Coppo, M. Dezani-Ciancaglini, S. Drossopoulou, and E. Giachino. Amalgamating Sessions and Methods in Object Oriented Languages with Generics. *Theoretical Computer Science*, 2008. To appear.

[CDCY07] M. Coppo, M. Dezani-Ciancaglini, and N. Yoshida. Asynchronous Session Types and Progress for Object-Oriented Languages. In *FMOODS'07*, *LNCS* 4468: 1–31. Springer-Verlag, 2007.

[CHY] M. Carbone, K. Honda, and N. Yoshida. Web Services, Mobile Processes and Types. *EATCS Bulletin* 91: 160–188, 2007.

[CHY07] M. Carbone, K. Honda, and N. Yoshida. Structured Communication-Centred Programming for Web Services. In *ESOP'07*, *LNCS* 4421: 2–17. Springer-Verlag, 2007.

[CR05] S. Crafa and S. Rossi. A Theory of Non-interference for the π-calculus. In *TGC'05*, *LNCS* 3705: 2–18. Springer-Verlag, 2005.

[DCDGY07] M. Dezani-Ciancaglini, S. Drossopoulou, E. Giachino, and N. Yoshida. Bounded Session Types for Object-Oriented Languages. In *FMCO'06*, *LNCS* 4709: 207–245. Springer-Verlag, 2007.

[DCdLY08] M. Dezani-Ciancaglini, U. de' Liguoro, and N. Yoshida. On Progress for Structured Communications. In *TGC'07*, *LNCS* 4912: 257–275. Springer-Verlag, 2008.

[DCMYD06] M. Dezani-Ciancaglini, D. Mostrous, N. Yoshida, and S. Drossopoulou. Session Types for Object-Oriented Languages. In *ECOOP'06*, *LNCS* 4067: 328–352. Springer-Verlag, 2006.

[FG01] R. Focardi and R. Gorrieri. Classification of Security Properties (Part I: Information Flow). In *Foundations of Security Analysis and Design - Tutorial Lectures*, *LNCS* 2171: 331–396. Springer-Verlag, 2001.

[GCDC06] P. Garralda, A. Compagnoni, and M. Dezani-Ciancaglini. BASS: Boxed Ambients with Safe Sessions. In *PPDP'06*, 61–72. ACM Press, 2006.

[GM82] J. A. Goguen and J. Meseguer. Security Policy and Security Models. In *IEEE Symposium on Secrecy and Privacy*, 11–20, 1982.

[Hen04] M. Hennessy. The Security π-calculus and Non-interference. *Journal of Logic and Algebraic Programming* 63(1): 3-34, 2004.

[HR02] M. Hennessy and J. Riely. Information Flow vs Resource Access in the Asynchronous π-calculus. *ACM TOPLAS* 24(5): 566-591, 2002.

[HVK98] K. Honda, V. Vasconcelos, and M. Kubo. Language Primitives and Type Disciplines for Structured Communication-based Programming. In *ESOP'98*, *LNCS* 1381: 22–138. Springer-Verlag, 1998.

[HVY00] K. Honda, V. Vasconcelos and N. Yoshida. Secure Information Flow as Typed Process Behavior. In *ESOP'00*, *LNCS* 1782: 180–199. Springer-Verlag, 2000.

[HY07] K. Honda and N. Yoshida. A Uniform Type Structure for Secure Information Flow. *ACM TOPLAS* 29(6), 2007.

[HYC08] K. Honda, N. Yoshida, and M. Carbone. Multiparty Asynchronous Session Types. In *POPL'08*, 273–284. ACM Press, 2008.

[Kob05] N. Kobayashi. Type-based Information Flow Analysis for the Pi-Calculus. *Acta Informatica* 42(4-5): 291-347, 2005.

[Kol08] M. Kolundzija. Security Types for Sessions and Pipelines. In *WS-FM'08*, *LNCS*. Springer-Verlag, 2008. To appear.

[Pot02] F. Pottier. A Simple View of Type-Secure Information Flow in the π-Calculus. In *CSFW'02*, 320–330. IEEE Computer Society, 2002.

[RS99] P. Ryan and S. Schneider, Process Algebra and Non-interference. In *CSFW'99*, 214–227. IEEE Computer Society, 1999.

[SM03] A. Sabelfeld and A. C. Myers, Language-based Information-flow Security. *IEEE Journal on Selected Areas in Communications* 211:5-19, 2003.

[THK94] K. Takeuchi, K. Honda, and M. Kubo. An Interaction-based Language and its Typing System. In *PARLE'94*, *LNCS* 817: 398–413. Springer-Verlag, 1994.

[VGR06] V. Vasconcelos, S. Gay, and A. Ravara. Typechecking a Multithreaded Functional Language with Session Types. *Theorical Computer Science*, 368(1-2): 64–87, 2006.

[VRG04] V. Vasconcelos, A. Ravara, and S. Gay. Session Types for Functional Multithreading. In *CONCUR'04*, *LNCS* 3170: 497–511. Springer-Verlag, 2004.

[YV07] N. Yoshida and V. Vasconcelos. Language Primitives and Type Disciplines for Structured Communication-based Programming Revisited. In *SecRet'06*, *ENTCS* 171: 73–93. Elsevier, 2007.