Scientific Report

First name / Family name	Demis / Ballis
Nationality	Italian
Name of the *Host Organisation*	Technical University of Madrid
First Name / family name of the *Scientific Coordinator*	Manuel / Carro
Period of the fellowship	01/02/2013 to 31/01/2014

I – SCIENTIFIC ACTIVITY DURING YOUR FELLOWSHIP

In recent years, the automated verification of web applications has become a major field of research. Nowadays, a number of corporations interact primarily through the web by means of web applications that combine static content with dynamic data produced "on-the-fly" by the execution of web scripts (e.g. Java servlets, Microsoft ASP.NET and PHP code). The inherent complexity of such highly concurrent systems has turned their verification into a challenge.

The focus of my research activity was on the use of formal methods (mainly based on the rewriting logic formalism) to build an integrated framework for the rigorous specification and verification of web applications. A relevant part of my research activity has been carried out in collaboration with Prof. María Alpuente and her group of the Technical University of Valencia (SpaRCIM). Below, I describe my work in accordance with the ABCDE Research Training Programme originally submitted to ERCIM.

TASK T1

Activity: We specified a formal, fine-grained web application model that accurately describes the behaviour of real-size web applications and that is suitable for the verification of the core business logic of complex, dynamic web systems. The model considers several critical aspects of concurrent Web interactions such as HTTP communication, forward/backward navigation, page refresh, and new window/tab opening. The formalization has been specified using the rewriting logic framework by means of suitable rewrite theories.
Outcome: The main results of Task T1 activity have been published in [1].

TASK T2

Activity: We developed a methodology to formally verify web applications. More specifically, we applied model-checking techniques to formally verify several important classes of properties (e.g., reachability, safety, authentication, mutual exclusion, liveness, and fairness conditions) w.r.t. web applications specified in accordance with the formal model defined in Task T1.
Outcome: The main results of Task T2 activity have been published in [1] and [3].

TASK T3

Activity: We studied the synergy among trace slicing, model-checking and automated debugging of rewriting logic theories to define an integrated framework in which erroneous web application components can be automatically detected, and explanations of the recognized anomalies can be synthesized from counterexamples.
More concretely, we developed forward and backward trace slicing techniques which can applied to counterexamples generated by model-checking faulty Web applications. The advantage of this approach is twofold. On the one hand, trace slicing greatly reduces the size of counterexamples favoring their inspection for debugging and analysis purposes. On the other hand, relevant information (e.g., control/data dependence and causality) can be recognized, exposing opportunities for program optimization.

Outcome: The main results of Task T3 activity have been published in [1,2,4,5,6]. The article [7] has been submitted to the Journal of Symbolic Computation.

TASK T4

Activity: We implemented several prototypical systems and conducted experimental evaluations to ascertain the usefulness of the proposed methodologies.
Outcome:

- **Web-TLR** is a software system designed for specifying and model-checking Web applications which is based on rewriting logic. Web applications are expressed as rewrite theories which can be formally verified by using the Maude built-in LTLR model-checker. Web-TLR is equipped with a user-friendly, graphical Web interface that shields the user from unnecessary information. Whenever a property is refuted, an interactive slideshow is generated that allows the user to visually reproduce, step by step, the erroneous navigation trace that underlies the failing model checking computation. **Web-TLR** is freely available at the URL `http://zenon.dsic.upv.es:8080/webtlr/checker.html`

- **iJulienne** is a trace analyzer for conditional rewriting logic theories which is based on backward trace slicing. It can be used to compute abstract views of computations that help users understand and debug rewriting logic specifications. The core engine of **iJulienne** has been integrated into **Web-TLR** to track back reverse dependencies and causality along web application counter-example traces. **iJulienne** is freely available at `http://safe-tools.dsic.upv.es/iJulienne/`

- **Anima** is a software tool that implements a trace inspection technique for rewriting logic specification that allows the non-deterministic execution of a given unconditional rewrite theory to be followed up in different ways. By selecting different inspection criteria, one can automatically derive a family of practical algorithms such as program steppers and more sophisticated dynamic forward trace slicers that facilitate the dynamic detection of control and data dependencies across the program computations. **Anima** is freely available at `http://safe-tools.dsic.upv.es/anima/`

II – PUBLICATION(S) DURING YOUR FELLOWSHIP

REFEREED SCIENTIFIC JOURNALS

[1] M. Alpuente, D. Ballis, and D. Romero. A Rewriting Logic Approach to the Formal Specification and Verification of Web applications. Science of Computer Programming. Elsevier, 2013 (in press).

[2] M. Alpuente, D. Ballis, F. Frechina, and D. Romero. Using Conditional Trace Slicing for Improving Maude Programs. Science of Computer Programming, volume 80, part B, pages 385-415. Elsevier, 2013.

[3] M. Alpuente, D. Ballis, M. Falaschi, F. Frechina, and D. Romero. *Rewriting-based repairing strategies for XML repositories*. The Journal of Logic and Algebraic Programming, Volume 82, Issue 8, pages 326-352. Elsevier, 2013.

REFEREED INTERNATIONAL CONFERENCES

[4] M. Alpuente, D. Ballis, F. Frechina, and J. Sapiña. *Inspecting Rewriting Logic Computations (in a parametric and stepwise way)*. In Specification, Algebra, and Software. A Festschrift Symposium in Honor of Kokichi Futatsugi (SAS 2014). Springer LNCS (to appear).

[5] M. Alpuente, D. Ballis, F. Frechina, and J. Sapiña. *Slicing-based Trace Analysis of Rewriting Logic Specifications with iJulienne*. In 22nd European Symposium on Programming (ESOP 2013), pages 121-124, Rome (Italy), 2013. Springer LNCS 7792, 2013.

[6] M. Alpuente, D. Ballis, F. Frechina, and J. Sapiña. *Parametric Exploration of Rewriting Logic Computations*. In 5th Int'l Symposium on Symbolic Computation in Software Science (SCSS 2013), Hagenberg (Austria), 2013. RISC-Linz Report Series No. 13-06, 2013.

SUBMITTED ARTICLES

[7] M. Alpuente, D. Ballis, F. Frechina, and J. Sapiña. *Forward Exploration of Maude Computations*. Submitted to Journal of Symbolic Computation, 2014.

III – ATTENDED SEMINARS, WORKHOPS, CONFERENCES

- 5th Int'l Symposium on Symbolic Computation in Software Science (SCSS 2013), July 5-6, 2013, Hagenberg (Austria),.
- ABCDE Seminar III, October 31 – November 1, 2013, Athens (Greece).

IV – RESEARCH EXCHANGE PROGRAMME (REP)

- **INRIA Sophia Antipolis, September 22-27, 2013**

  Activity: During my stay at INRIA, I collaborated with Dr. Luigi Liquori on the verification of trust and reputation systems and their related protocol specifications.
  Specifically, we analyzed some existing trust and reputation systems for online marketplaces and shared ideas on some possible formalizations of these systems using the rewriting logic framework and the Maude formal environment.

- **University of Málaga (SpaRCIM), November 25-29, 2013**

  Activity: During my stay at the University of Málaga, I gave a lecture on the formal specification and verification of web applications, and I also started a scientific collaboration with Prof. María del Mar Gallardo and her group on the formal specification and synthesis of mission-critical hybrid systems.
  In particular, we studied the possibility to synthesize dam controllers for flood management using RTMaude, the real-time extension of the Maude specification language.