# Enhanced Graph Rewriting Systems for Complex Software Domains (SoSyM Abstract)

Cédric Eichler, Thierry Monteil, Patricia Stolf, Alfredo Grieco, Khalil Drira

## HAL Id: hal-01228299

## https://hal.archives-ouvertes.fr/hal-01228299

Submitted on 8 Jan 2016

# Enhanced Graph Rewriting Systems for Complex Software Domains

## - Dynamic Software Architecture, Non-Functional Requirements And Correctness By Construction (SoSyM Astract)

Cédric Eichler [*†‡], Thierry Monteil[*‡], Patricia Stolf[†‡], Alfredo Grieco[§] and Khalil Drira[*‡]

[*]CNRS, LAAS 7 avenue du colonel Roche, F-31400 Toulouse, France
Email: author_name@laas.fr
[†]IRIT; 118 Route de Narbonne, F-31062 Toulouse, France
Email: author_name@irit.fr
[‡] Univ de Toulouse, UPS F-31400, INSA F-31400,
UTJ, F-31100 Toulouse, France
[§]Department of Electrical and Information Engineering
Politecnico di Bari,
Via Orabona 4 - 70125, Bari, Italy

*Abstract*—Methodologies for *correct by construction reconfigurations* can efficiently solve consistency issues in dynamic software architecture. Graph-based models are appropriate for designing such architectures and methods. At the same time, they may be unfit to characterize a system from a non functional perspective. This stems from efficiency and applicability limitations in handling time-varying characteristics and their related dependencies. In order to lift these restrictions, an extension to graph rewriting systems is proposed herein. The suitability of this approach, as well as the restraints of currently available ones, are illustrated, analysed and experimentally evaluated with reference to a concrete example. This investigation demonstrates that the conceived solution can: (i) express any kind of algebraic dependencies between evolving requirements and properties; (ii) significantly ameliorate the efficiency and scalability of system modifications with respect to classic methodologies; (iii) provide an efficient access to attribute values; (iv) be fruitfully exploited in software management systems; (v) guarantee theoretical properties of a grammar, like its termination.

This is an extended abstract for the Models 2015 Conference of the journal paper of the same name [1].

## I. Motivation

Dynamic software architectures enable adaptation in evolving distributed systems. A crucial undesirable implication of such adaptations is a potential loss of correctness, the system withdrawing from its scope of consistency.

Besides correctness, the system has evolving non-functional requirements, which are tightly linked to its appropriateness or efficiency. The satisfaction of these *objectives* depends on the *properties* of the system, its components, and their relations.

On one hand, graph-based models are appropriate for the design of adaptation rules that necessarily preserve the system's consistency. On the other, currently available graph based methods exhibit limitations with regard to the description of system properties, in particular regarding their evolution and inter-dependencies.

## II. Proposed solution

A formal extension of graph rewriting systems is then proposed to lift these shortcomings. Inspired from classical string rewriting theory, the pivotal features of this enhancement are: *mutators*, *admissible relationships* specification, and *constraint oriented* encoding.

## III. Results

It is demonstrated that the proposed solution is appropriate for the management of systems subordinate to functional and non-functional requirements:

Experimental results show that reconfiguring a graph of size 2500 with the introduced method rather than existing ones is up to 27 times quicker on GMTE and 8.3 times quicker on AGG.

This improvement allows to efficiently asses characteristics of the system by combining evaluation on demand and/or update on modification.

In turn, this allows to quickly grasp the appropriateness of a configuration, identify objectives that can be ameliorated, and component implying constraints violation.

## References

[1] C. Eichler, T. Monteil, P. Stolf, L. Grieco, and K. Drira, "Enhanced graph rewriting systems for complex software domains," *Software & Systems Modeling*, pp. 1–21, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10270-014-0433-1