The FMICS-jETI Platform: Status and Perspectives

Tiziana Margaria  
Chair of Service and Software Engineering  
Universität Potsdam  
August Bebel Str. 89, 14482 Potsdam, Germany  
margaria@cs.uni-potsdam.de

Christian Kubczak  
Chair of Software Engineering  
Universität Dortmund  
Otto-Hahn-Str. 14, 44227 Dortmund, Germany  
christian.kubczak@cs.uni-dortmund.de

Bernhard Steffen,  
Chair of Programming Systems  
Universität Dortmund  
Otto-Hahn-Str. 14, 44227 Dortmund, Germany  
steffen@cs.uni-dortmund.de

Stefan Naujokat  
Chair of Programming Systems  
Universität Dortmund  
Otto-Hahn-Str. 14, 44227 Dortmund, Germany  
stefan.naujokat@cs.uni-dortmund.de

Abstract

One of the goals of FMICS, the ERCIM Working Group on Formal Methods for Industrial Critical Systems (FMICS) [8], is to transfer and promote the use formal methods technology in industry. The ongoing Verified Software Repository Grand Challenge [11] offers a great opportunity to reach this goal, resulting in a more robust and solid software industry in Europe. We demonstrate here the current status of the FMICS-jETI platform, a collaborative demonstrator based on the jETI technology, that provides as repository a collection of verification tools stemming from the activities of the FMICS working group and facilities to orchestrate them in a remote and simple way. At the same time FMICS-jETI itself is a contribution to the VSR repository and thus to the Grand Challenge.

1 Introduction

The FMICS community develops since its inception methods, tools, and their applications to industrial critical systems. Our point of view thus stresses correctness at the model level, rather than at the software (more generally at the coding) level. An adequate repository should therefore contain not only analyzed or proven correct software, but principally tools (themselves software artifacts) that help establishing the correctness of the software in question starting from the requirements, specifications, and models. We are convinced that this perspective on modelling tools and even frameworks can be useful for the progress towards better software also for less critical application domains, like consumer IT products: they strike wherever time to market and first time correct are an issue.

UK has initiated effort towards a verified software repository [5] within the framework of the UK Grand Challenges Programme [10]. The Verified Software Repository will assist in the development of software by facilitating access to a managed collection of analysis tools and a repository of case studies or challenge codes to exercise these tools. The emphasis will be on flexibility: the potential to encompass a broad range of analysis techniques, such as verifiers, theorem provers, model checkers, static analyzers, test case generators, etc and a broad range of design artefacts, such as documents, codes, test suites, safety cases, etc. As a result, the repository will make it easier for software engineers to learn how to use analysis tools effectively. It will also provide examples of good practice for software engineers and facilitate further development and improvement of both the tools and the examples, by bringing them together with common standards.

2 The FMICS Contribution

Based on jETI [19], the new generation of ETI [18, 1], the core FMICS partners have set up a collaborative demonstrator that

- illustrates the applicability of the jETI technology for lightweight remote integration of tools into the repository
- shows how to provide tools to the repository, by registration and remote provision,
The jETI Tool Integration Workflow demonstrates how to experiment with local and remote tools to solve cooperative verification tasks. It shows how to orchestrate different tools (possibly a mix of local and remote ones) which were not originally designed to cooperate, to address more complex case studies. This may require the availability of mediators, to cover semantic gaps between the tools.

The participating partners are so far CCLRC/RAL (UK), CWI (The Netherlands), INRIA Grenoble (France), ISTI Pisa (Italy), Masaryk University in Brno (Czech Republic), University of Dortmund (Germany), the University of Malaga (Spain), University of Potsdam (Germany), University of Saarbrücken (Germany), as shown online (see Fig. 2).

The core effort by the partners concerns the provision of tools and case studies that serve as executable benchmarks of the demonstrator. The coordinators (Univ. of Dortmund and Univ. of Potsdam) are responsible for the jETI platform and technology, and for the development and test of the demonstration between their two locations.

Early results have been shown at the FMICS Session of the IDPT Conference 2006 [3] and at the VSTTE Dagstuhl Workshop in July 2006.

### 3 Easy Remote Tool Integration in jETI

jETI's purpose is unique in

- allowing users to combine functionalities of tools of different application domains to solve problems a single tool never would be able to tackle and
- following a service-oriented approach that combines heterogeneous services provisioned in different technologies. They can be Web services, like in the bioinformatics domain [16] and in the ongoing Semantic Web Service Challenge [13, 21], but also so-called REST services, which is still the most widespread case on the World wide web and which is also the FMICS normal case.

REST (REpresentational State Transfer) services [7] deal in a structured way with resources over the Web, without prescribing the adhesion to a specific technology (like Webservices), but prescribing adhesion to a description of the location of the service, of its signature, and a restriction to stateless interactions.

Obviously, the richness of the tool repository plays a crucial role in the success of jETI-FMICS: the benefit gained from our experimentation and coordination facilities grows with the amount and variety of integrated software-tools.
The success of the jETI concept is thus highly sensitive to the process and costs of tool integration and tool maintenance.

Taking advantage of newer technologies that internally base on Web Services and Java technology, we are

1. considerably simplifying the integration process, and at the same time
2. flexibilizing the distribution, version management and use of integrated tools,
3. broadening the scope of potential user profiles and roles, by seamlessly integrating ETI's coordination and synthesis features (cf. [18]) with a standard Java development environment, and
4. solving the scalability problem connected with tool maintenance and evolution.

The background and a first attempt to the new distributed way of tool integration for ETI have been described in [15]. Our current version of ETI, jETI,

- exploits Web services technology [23, 24, 20] to further simplify the remote tool integration and execution, but it is not tied to Web services
- supports cross platform execution of the coordination models based on the quasi standard set by Java, and it naturally
- flexibilizes the original coordination level by seamlessly integrating the Eclipse development framework.

Altogether, the architecture for the FMICS-jETI platform is a typical SOA/SOC setting, as shown in Fig. 3 that additionally covers REST services.

### 3.1 Registration instead of Integration

jETI’s integration philosophy addresses the major obstacle for a wider adoption, as identified during seven years of experience with tool providers, tool users and students: the difficulty to provide the latest versions of the state-of-the-art tools. The tool integration process required on dedicated ETI servers was too complicated for both the tool providers and the ETI team, making it impossible to keep pace with
the development of new versions and a wealth of new tools. jETI’s new remote integration philosophy overcomes this problem, because it replaces the requirement of ‘physical’ tool integration by very simple registration and publishing. This allows the provisioning of tool functionalities in a matter of minutes: fast enough to be fully demonstrated during our presentation. Moreover, whenever the portion of a tool’s API which is relevant for a new version of a functionality remains unchanged, version updating is fully automatic!

Based on the Web Services functionality, the realization of this registration/publishing-based integration philosophy required the implementation of four components, as illustrated in Figure 1:

1. a **HTML Tool Configurator**, which allows tool providers to register a new tool functionality just by filling out a simple template form,

2. the **jABC Component Server**, which (a) automatically generates appropriate Java classes from these specifications and (b) organizes all the registered tool functionalities, including the corresponding version control,

3. the **jETI client**, which automatically loads the relevant Java classes from the jABC Component Server and provides a flexible Java development environment for coordinating the so obtained tool functionalities. Depending on their goals and skill profile, users may just use our graphical coordination editor to experiment with the tools, or they may use the full development support of Eclipse to really embed remote functionalities into normal Java programs. Of course, this choice heavily influences the size of the required jETI-Client: only the first option is open to our envisaged ‘pure HTML’ solution.

4. a **Tool Executor**, which is able to steer the execution of the specified tools at the tool providers’ site.

This approach enables **experts** to develop complex tools in Java on the basis of a library of remotely accessible tool functionalities, as well as **newcomers** to use jETI’s formal methods-based, graphical coordination environment to safely combine adequate tool functionalities into heterogeneous tools.
4 Status of jETI-FMICS

So far, the technology has been used to integrate jMoSel [22, 12], a decision procedure for the verification of parametric systems, DFA-MC [14], a program analysis tool that carries out dataflow analyses of (Java) programs via model checking, and GEAR [4, 9], our game-based model checker for modal mu-calculus, as shown in the workflow in Fig. 4.

The group at University of Malaga is currently integrating the SPIN [17] tool for LTL model checking and αSPIN [2] a tool for the definition of sound model abstractions. In parallel there is work on the integration at different levels of the CADP toolsuite [6], which includes itself a number of tools and transformers/optimizers as well as the capability to define compound tool executions. Here it is particularly interesting to see how to exploit the orchestration techniques of CADP and of jETI for different kinds of users.

Acknowledgement

We thank Ralf Nagel and Stephan Windmüller for their contribution to the jETI environment and to establishing the FMICS-jETI platform.

References


