Cloud Application Modelling and Execution Language (CAMEL) and the PaaSage Workflow

Alessandro Rossini
SINTEF, Oslo, Norway
alessandro.rossini@sintef.no

Model-driven engineering (MDE) is a branch of software engineering that aims at improving the productivity, quality, and cost-effectiveness of software development by promoting models and model transformations as the primary assets in software development. Models can be specified using general-purpose languages like the Unified Modeling Language (UML). However, to fully unfold the potential of MDE, models are frequently specified using domain-specific languages (DSLs), which are tailored to a specific domain of concern.

The PaaSage project delivers a platform to support the modelling, execution, and adaptation of multi-cloud applications (i.e., applications deployed across multiple private, public, or hybrid cloud infrastructures). In order to cover the necessary aspects of the modelling and execution of multi-cloud applications, PaaSage adopts the Cloud Application Modelling and Execution Language (CAMEL) [5].

CAMEL integrates and extends existing DSLs, namely the Cloud Modelling Language (CloudML) [1], Saloon [4], and the Organisation part of CERIF [2]. In addition, CAMEL integrates new DSLs developed within the project, such as the Scalability Rule Language (SRL) [3].

CAMEL enables PaaSage users to specify multiple aspects of multi-cloud applications, such as provisioning and deployment topology, provisioning and deployment requirements, service-level objectives, metrics, scalability rules, providers, organisations, users, roles, security controls, execution contexts, execution histories, etc.

In order to facilitate the integration across the components managing the life-cycle of multi-cloud applications, PaaSage leverages upon CAMEL models that are progressively refined throughout the modelling, deployment, and execution phases of the PaaSage workflow (see Figure 1):

- **Modelling phase**: The PaaSage users design a cloud-provider independent model (CPIM), which specifies the deployment of a multi-cloud application along with its requirements and objectives in a cloud provider-independent way.
- **Deployment phase**: The Profiler component consumes the CPIM, matches this model with the profile of cloud providers, and produces a constraint problem. The Reasoner component solves the constraint problem (if possible) and produces a cloud-provider specific model (CPSM), which specifies the deployment of a multi-cloud application along with its requirements and objectives in a cloud provider-specific way. The Adapter component consumes the CPSM and produces deployment plans, which specify platform-specific details of the deployment.

1. [http://www.paasage.eu](http://www.paasage.eu)
– **Execution phase**: The Executionware consumes the deployment plans and enacts the deployment of the application components on suitable cloud infrastructures. Finally, the Executionware records historical data about the application execution, which allows the Reasoner to look at the performance of previous CPSMs when producing a new one.

By leveraging upon CAMEL models not only at design-time but also run-time, PaaSage enables self-adaptive multi-cloud applications (*i.e.*, multi-cloud applications that automatically adapt to changes in the environment).

**Acknowledgements.** The research leading to these results has received funding from the European Commission’s Seventh Framework Programme (FP7/2007-2013) under grant agreement number 317715 (PaaSage).

**References**