

# Morphemic Cloud Application Models Design

Alessandra Bagnato  
Softeam Software dept.  
Softeam (Docaposte Group)  
Paris, France  
alessandra.bagnato@softeam.fr

Etienne Brosse  
Softeam Software dept.  
Softeam (Docaposte Group)  
Paris, France  
etienne.brosse@softeam.fr

Kaïs Chaabouni  
Softeam Software dept.  
Softeam (Docaposte Group)  
Paris, France  
kais.chaabouni@softeam.fr

**Abstract**— The Morphemic H2020 project covers several features from modelling cross-cloud applications, continuous and autonomous optimization and deployment and providing access to several cloud capabilities. This demo paper describes the MORPHEMIC CAMEL Designer tool responsible of the Cloud Application Modelling and Execution Language (CAMEL) design for the modelling Environment Modelio. CAMEL Designer is an open source module for graphically creating, editing and exporting CAMEL Models in XMI format.

**Keywords**— Cloud, Fog, edge and multi-access computing and networking, Open-source projects, open standards.

## I. INTRODUCTION

Modelling and orchestrating heterogeneous resources and polymorphic applications for holistic execution and adaptation of models in the cloud, the Morphemic Project [1] covers areas including cloud computing; big data and open data; and artificial intelligence. 12 partners from 7 countries are developing the innovative MORPHEMIC platform, which will provide a unique way of adapting and optimizing cloud computing applications for future specialized hardware configurations like GPUs, TPUs, AI chips, FPGA, HPC. MORPHEMIC is an extension of the MELODIC multi-cloud platform and is a single universal platform that facilitates and optimizes deployment and management of applications cross-cloud [2].

MORPHEMIC introduces two novel concepts to cloud computing which will utilize cloud computing resources in the most optimal way with the goal to simplify Cloud application modelling.

Polymorphing architecture will allow for dynamic adaptation of the architecture of application to the current workload. When a component can run in different technical forms (i.e. in a virtual machine (VM), in a container, as a big data job, or as serverless components, etc.), depending on the application's requirements and its current workload, its components could be deployed in various forms and in different environments. This allows to maximize the specified utility function of the application deployment such as optimizing deployment cost, and to satisfy the specified constraints such as the accepted response time or the resource requirements. Thus, this will change the component type (VM, container, serverless, GPU or FPGA) based on the optimal found solution.

Proactive adaptation is simply a proactive approach that allows the reconfiguring of the application based on

forecasted metrics about usage and workload level, in order to adapt the application in the most optimal way by predicting the violation of the requirements as opposed to reactive approach that is reconfiguring the application after the violation of the requirements. Therefore, the objective is to forecast future resource needs and possible deployment configurations. This ensures that adaptation can be done effectively and seamlessly for the application users.

Section II of the paper describes the camel designer module while Sections II – VIII describe the various steps of our demo.

## II. CAMEL DESIGNER MODULE

CAMEL Designer [3] is an open source module for graphically creating, editing, and exporting CAMEL Models in an ergonomic, intuitive, and user-friendly environment. CAMEL Designer module is used via Modelio [4] tool community has all the necessary tools for creating the CAMEL model and all its sub models and components.

Modelio [4] is a modelling tool for many standards for System Engineering, Software Development and Enterprise Architecture. For instance, Modelio includes the standard Business Process Model and Notation (BPMN) for modelling business processes, the standard Unified Modeling Language (UML) and the ArchiMate language for Enterprise Architecture. Modelio offers services and diagrams for modelling and can assist with model auditing and consistency verification. Furthermore, for additional functionalities that require the extension of Modelio modelling capabilities or extending its metamodels, the Modelio Module API is used to create a new module to be deployed into a Modelio project.

Fig. 1 illustrates how CAMEL Designer is implemented as module to be deployed on Modelio. CAMEL Designer module includes a UML profile for the CAMEL Language which is implemented by associating stereotypes based on CAMEL concepts to different UML meta-classes such as “Class”, “Package”, “Attribute”, etc. These elements of the CAMEL Profile are created by additional “Commands” provided by CAMEL Designer module. In addition to these commands, the user can design graphically with the diagram tools and with the help of a property page widget for accessing and editing model elements properties. Further details about the modelling environment of CAMEL Designer are provided in the next section. Moreover, CAMEL Designer invokes the CAMEL DSL API for parsing CAMEL files during import and serializing CAMEL files during export.

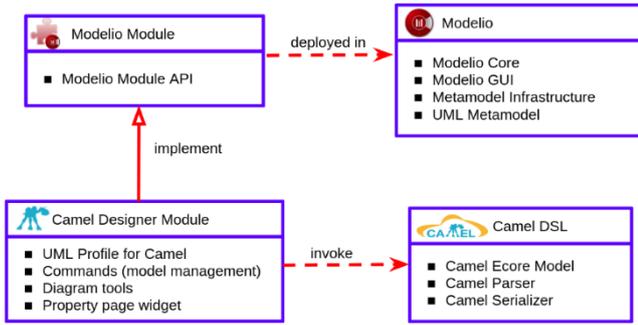


Fig. 1: CAMEL Designer Module and Modelio Overview

Fig. 2 illustrates the modelling environment provided by Modelio which contains a model explorer depicted on the left side of Fig. 2. This model explorer shows the hierarchy of the persisted model elements and allows to create, delete and copy/paste other model elements. On the right side of Fig. 2, the diagram represents how the elements in the model are represented and linked together. In addition, a set of tools is provided for each diagram to allow the user to modify the model such as adding new elements, properties, dependencies or just customizing the visual appearance of the elements illustrated in the diagram. Moreover, other commands can be called directly on elements from the model explorer by right clicking the selected element and executing the command which may bring changes to the model. Finally, on the bottom of Fig. 2, the properties section displays a list of properties of the selected element and allows the user to edit these properties.

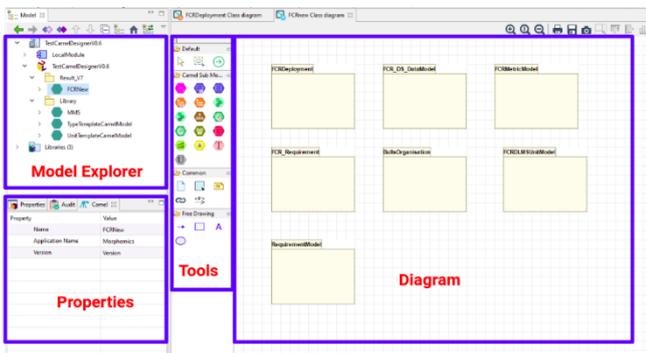


Fig. 2: Modelio Environment for CAMEL Designer

### III. CAMEL MODELS MANAGEMENT

The first step of starting with CAMEL Designer is to create and open a Modelio project and deploy CAMEL Designer module into the project. To create a new CAMEL Model, simply right click a package where you wish to create your CAMEL model and select the command “Create a new CAMEL Model” (see Fig. 3). In a similar way, this applies to importing existing CAMEL Models files on CAMEL or XMI format. As a result, a new CAMEL Model is created, and a new diagram is opened to edit the newly created CAMEL Model.

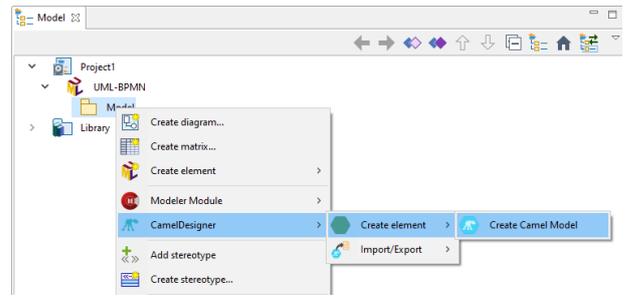


Fig. 3: Create a CAMEL Model Command

After having successfully created a CAMEL model, you can then create sub-models on CAMEL model root by clicking on the tools displayed on the palette tools on the left of the CAMEL Model diagram. For example, Fig. 4 shows the palette tools on the left where it is used to create the Requirement Model and the Type Model.

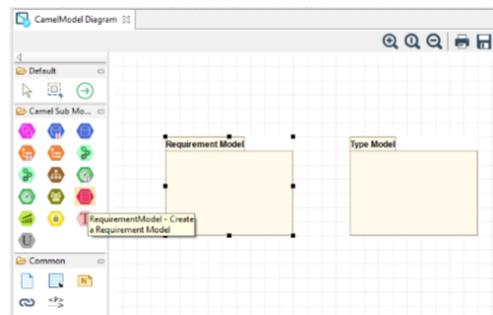


Fig. 4: CAMEL Model Diagram Tools

### IV. CAMEL DEPLOYMENT SPECIFICATION

The first step of starting with CAMEL Designer is to create and open Modelio/Camel project with a CAMEL model containing a Deployment model and an associated Deployment diagram. Once the CAMEL model is initialized, the user can create a Deployment model and diagram as depicted in Fig. 5. By using the tools, as shown in Fig. 5, the deployment model can be enriched with dedicated deployment concepts.



Fig.5: Simple CAMEL Component deployment

### V. CAMEL REQUIREMENT SPECIFICATION

The user can create a Requirement model and diagram as depicted in Fig. 6. By using the tools provided by the CAMEL Requirement diagram, the user can specify the CAMEL concepts related to its Requirement aspects. These aspects could include the definition of several requirements, their grouping inside a “RequirementSet” and finally their reference by a software component for example, as depicted in Fig. 6.

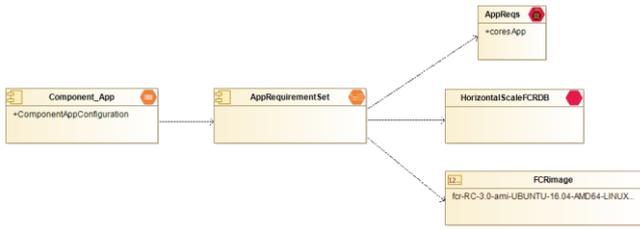


Fig.6: Example of requirement specification

## VI. CAMEL METRIC SPECIFICATION

To create an Optimisation specification, several CAMEL concepts need to be specified. First, information needs to be defined in order to be collected. For example, Fig. 7 depicts the definition of a “Template” named CardinalityTemplate. This latter is based on a cardinality “Measurable Attribute” and is a positive integer (ZeroToHundredInteger) representing the number of instances.

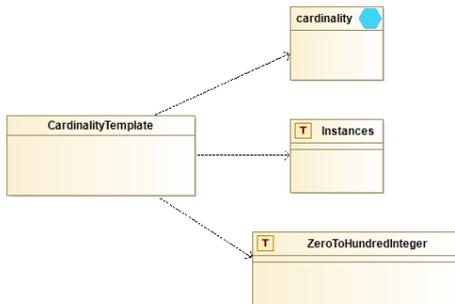


Fig. 7: CardinalityTemplate

Then an AppActCardinality “Variable”, see Fig. 8, extends the predefined CardinalityTemplate “Template”, is applied to the Component-App “Software Component” to count the number of instances of this particular software component.

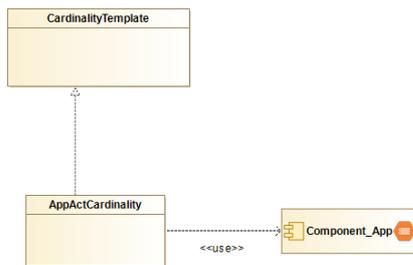


Fig. 8: AppCardinality “Variable” definition

The CAMEL designer provides also diagrams to specify “Locations” as depicted in Fig. 9.

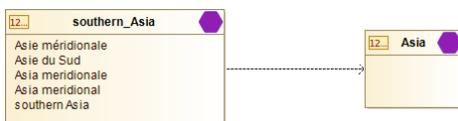


Fig. 9: CAMEL Location specification

## VII. CAMEL MODELS IMPORT/EXPORT

After completing the design of the CAMEL Models, the model can be exported in CAMEL or XMI format, by right clicking on the model explorer and use the command “Export CAMEL Model” (see Fig. 10). Then a dedicated window is opened to help to specify the name and the location of the exported file.

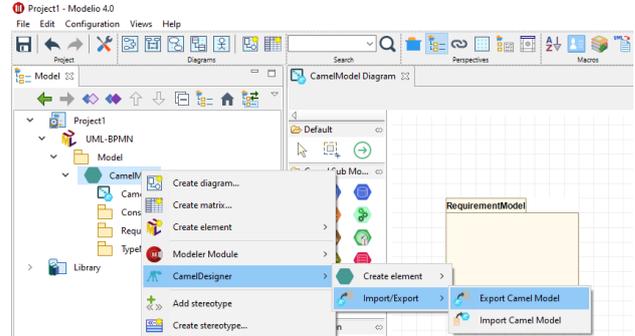


Fig. 10: Export CAMEL Mode

## VIII. CONCLUSIONS

The paper presented the CAMEL Designer module and its main features for the Modelio Modelling tool [4] at the current time of the MORPHEMIC project, the module can be downloaded under open source licence and it is available on the Modelio Research team GitHub page at [3]. The developed module will be applied in the next months to MORPHEMIC project’s use cases [1]. In particular ICON’s Computational Fluid Dynamics (CFD) products and expert services that support engineering analysis for a wide range of applications in the Automotive, Aerospace, Buildings, Health, Energy, Motorsport, Consumer Products and Space, IS-Wireless (ISW) use cases showing a 5G software defined base station and Lausanne University Hospital (CHUV) e-brain science and neuroimaging tools use cases will benefit using the CAMEL Designer described in this paper.

## ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 871643. The authors wish to thank all the MORPHEMIC Consortium members and Softeam Software team for their support.

## References

- [1] MORPHEMIC Web site, [www.morphemic.cloud](http://www.morphemic.cloud)
- [2] G. Horn and P. Skrzypek, "MELODIC: Utility Based Cross Cloud Deployment Optimisation," *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, doi: 10.1109/WAINA.2018.00112.
- [3] Modelio Camel Designer, [github.com/Modelio-R-D/CamelDesigner](https://github.com/Modelio-R-D/CamelDesigner)
- [4] Modelio Web Site, [www.modelio.org](http://www.modelio.org)

