

Morphemic Cloud Application Models Design

Alessandra Bagnato, Etienne Brosse and Kaïs Chaabouni Softeam Software dept. Softeam (Docaposte Group) SAAM Mobility 2021, 16th June 2021

Eclipse SAAM Mobility 2021 Security | Al | Architecture | Modelling

Supported by:

CLUSTER

AUTOMOTIVE



OULU

MADE AVAILABLE UNDER THE ECLIPSE PUBLIC LICENSE 2.0 (EPL-2.0)

Photo by Marc Sendra Martorell on Unsplash



MORPHEMIC projects aims to simplify **Cloud application modelling** and continuously optimize and morph the deployment model to take advantage of beneficial Cloud capabilities

Cloud Applications context

- 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.
- Cloud Applications can have variable resource demand
 - Reactive to changing execution context
 - Balance of cost performance experience



 MORPHEMIC will provide the ability to optimally adapt in a reactive manner the form/architecture and deployment plan of a given application based on the application requirements and the current context.

MORPHEMIC MORPHEMIC Consortium



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 Project https://www.morphemic.cloud/



- This talk 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

Cloud Application Modeling with the CAMEL Designer

- The MORPHEMIC CAMEL
 Designer tool is responsible of the
 Cloud Application Modelling and
 Execution Language (CAMEL)
 design for the modelling
 Environment Modelio.
- CAMEL Designer is an opensource module for graphically creating, editing and exporting CAMEL Models in XMI format.

MARQUE DE DOCAPOST

SOFT

Modelio





Modelio

Software and System Engineering

- UML editor with 25 years' history
 - Systems Engineering
 - Requirements,
 - SysML
 - Impact Analysis
 - Traceability features
 - MARTE, UTP2
 - TOGAF, BPMN
 - Code generation
 - Documentation
- Available under

open source at

<u>Modelio.org</u>

Commercial version at https://www.modeliosoft.com/en/



CxxProject - Modelio File Edit Configuration MDA Views Help - 👜 🙋 🚓 🏦 🗎 🏄 🖀 📓 🔊 🔊 Search: 📑 Modelio pers... - CxxProject 🛛 Diagrams 🛃 UML 🛛 □ ✿ ➡ ✿ ✿ ✿ ☆ 丹 摘 0 0 0 A B B DeploymentData GUI CxxProject Select CooProject Generation [] Marquee ITaskView CxxProject Compilation Class model CxxProject Doxygen docum Task manage... Windows C MyPlanner Interface P GUI 1015 M +Task +TaskWindows C: Windows Component ... + displayTitle : string [... Win32Types > Instance mode Task() EO + Attribute : CDC [1..1] TaskWindows A: + displayTitle : strin TaskWindows.cox TaskWindows.hxx A: +Attribute : CDC ↔ +task: Task [1] ED + TaskWindows() * File Type: Class body (0 + TaskWindows(va * Class: TaskWindows 00 + operator = (value */ DD +~TaskWindows() 00 + getDisplayTitle(): //class header file + getAttribute(): CE #include "MvPlanner/GUI/Windows/TaskWindows.hxx" 00 + getTask(): Task -> ITaskView (from GI _ 12 - 0 🔊 Audit 📑 Element 🛛 🔣 🖪 Annotations - D E Outline E Symbol C++ B Description of 'TaskWindows' 🎎 🔛 🖻 🖬 🖬 🖄 This class s used to graphically represent Task Value data. It renders Name TaskWindows Property Value - task information Visibility Public Class kind task status Abstract No code resources Primitive C++ na.... Is struct Active Summary Task visualization window Can be inherited Docume... This class s used to graphically repr. Root Main

www.morphemic.

Modelio Development Process (1/2)

- Create the new concepts
 - By extending UML Metamodel
- Requires

MORPHEMIC

- UML/Modelio Knowledge
- Concepts Metamodel/Examples
- Allow new concepts creation/modification
 - Commands for creation,
 - Views and tools for creation,
 - Property pages for editing.



Modelio Development Process (2/2)

- Create customs commands
 - Import/export

MORPHEMIC

- Facilitated by existing:
 - CAMEL DSL,
 - Examples.

| | | | > | () | Ecore |
|---------------|------------|--|-----|----|--|
| | | | > | | Model -> Feature |
| ≡ | 🖲 Bitbuck | et Projects Repositories - People - | > | | CamelModel -> Model |
| -1 | | | | | Action -> Feature |
| Tost | TESTS / | melodic | > | | Application -> Feature |
| . ↓ . | Sourc | e | ĺ, | ~ | LaverType |
| | 9 0 | and the (TestOreas (FOR (FOR | | _ | NewsedFlewsert |
| Î. | 17 mast | melodic / festCases / FCR / FCRnew.camer | > | | Namedelement |
| đ† | | | > | | Feature -> NamedElement |
| _ | Source | ce view Diff to previous History - Contributors - | > | | Attribute -> NamedElement |
| =G | 1 | camel model FCRnew{ | | | QualityAttribute -> Attribute |
| á | 2 | application FCRwithDLMSApp(| | Η | MeasurableAttribute -> QualityAttribute |
| | 3 | version '2.0' | í | | |
| | 5 | deployment type model FCRDeployment{ | × = | | epioyment |
| | 6 | software Component_App(requirements_AppRequirementSet | > | Ű |) Ecore |
| ¢ | 8 | required host WM AppHostReq | > | | DeploymentInstanceModel -> DeploymentModel |
| ь | 9 | <pre>script configuration ComponentAppConfiguration(</pre> | | | DeploymentModel -> Model |
| | 11 | configure 'mkdir ~/test2 && sudo ~/melodic/FCRAPP.sh install | | Η | DeploymentTypeModel -> DeploymentModel |
| \mathcal{V} | 12 | <pre>start 'printenv >> ~/melodic/env.txt && ~/melodic/FCRAPP.sh } }</pre> | | | Component & Fosture |
| 1 | 14 | | > | | Component -> Feature |
| | 15 | provided communication ComponentAppPort port 8087 required communication ComponentPortDbReq port 3306 mandatory | > | | SoftwareComponent -> Component |
| đ | 17 |) | | | VM -> Component |
| | 18 | software Component_LB { [MetaDataMode] MELODICMetadataSchema UtilituMotions UtilituRelat | > | | RequirementSet -> Feature |
| Ċ. | 20 | requirements LBRequirementSet | | | Configuration > Easture |
| | 21 | required host WM_LBHostReq | | | Configuration -> reature |
| | 22 | script configuration ComponentLBConfiguration{ | > | | Communication -> ComponentRelation |
| | 23 | download 'sudo service tomcat/ stop && rm -ri ~/load_balance | | | CommunicationDart > Feature |
| | 25 | configure 'printenv >> ~/load balancer/lb config env.txt 66 | > | | CommunicationPort -> reature |
| " | < | | | | ProvidedCommunication -> CommunicationPort |
| | | | > | | RequiredCommunication -> CommunicationPort |

platform:/resource/parent/camel/model/camel.ecore

✓ ⊕ camel
> i mport
> ⊕ mms
✓ ⊕ core

This project has received funding from the European Union's Horizon 2020 research and innovation programme and a start and a s



CAMEL Designer in the Modelio Modeling Tool

- A model explorer shows the hierarchy of the persisted model elements and allows to create, delete and copy/paste other model elements
- 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



Cloud Application Modeling with the CAMEL

Designer







MORPHEMIC Use Cases & Applications

- MORPHEMIC project & Modelio CAMEL Designer tools currently are aiming at offering the achieved results to all data and computation intensive organisations in need for the optimization of their existing cloud architecture and resources.
- The developed module & the whole Morphemic platform will be applied in the next months to MORPHEMIC project's use cases
 - 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.



- IS-Wireless has the ambition to deploy and adaptively provision its use-case application, exploiting 5G Software-defined Radio Access Networks (RAN), in cloud and hybrid (cloud & edge) environments.
- The generic requirements are the following:
 - the location of all components should be determined on a regional/country level granularity (e.g. deployment should be done in Poland)
 - each component has the requirements of 5 as minimum number of cores and 2 as the number of GBs for the RAM
 - the application average availability should be at least 99.999%





A Cloud RAN can comprise three main units on which the different protocols are distributed: the Radio Unit (RU) comprising low-level protocols, the **Distributed** Unit (DU) comprising intermediate-level protocols and the and the **Central Unit (CU)** comprising high-level protocols. The CU unit can be also separated into the control (CP) and user plane (UP) such that the respective parts, i.e., CU-CP and CU-UP can be independently managed and deployed. comprising high-level protocols.

Example (2/3)



The requirement model includes two link requirements covering the quality of communication between DU and CU-CP as well as between DU and CU-UP. Each communication requirement involves the specification of two attributes that define the respective constraints on communication latency and throughput.



Snippet of application's CAMEL model covering the requirement, metric and constraint domains, each component has the requirements of 5 as minimum number of cores and 2 as the number of GBs for the RAM

The communication-specific requirements concern the quality of the communication between pairs of components:

- The latency between DU and CU-CP should be at most 5 (milliseconds) while the throughput at least 0.1 Gbps for both directions of communication.
- The latency between DU and CU-UP should be at most 1 (millisecond) while the throughput at least 4 Gbps for both communication directions.

| ink r | requirement DU_CU_CP_COMREQ { | |
|-------|---|--|
| attr | ibute Throughput | |
| [Ⅳ | /letaDataModel.MELODICMetadataSchema.ApplicationPlacementModel.IaaS.Network | |
| En | itity.Network.NetworkQoS.hasTxput] : double 0.1 | |
| attr | ibute Latency | |
| [N | /etaDataModel.MELODICMetadataSchema.ApplicationPlacementModel.IaaS.Network | |
| En | tity.Network.NetworkQoS.hasLatency] : double 5.0 | |

attribute Throughput

[MetaDataModel.MELODICMetadataSchema.ApplicationPlacementModel.IaaS.Network Entity.Network.NetworkQoS.hasTxput]: double 4.0

attribute Latency

[MetaDataModel.MELODICMetadataSchema.ApplicationPlacementModel.laaS.Network Entity.Network.NetworkQoS.hasLatency] : double 1.0

resource requirement RES{

attribute RAM

[MetaDataModel.MELODICMetadataSchema.ApplicationPlacementModel.IaaS.Cloud.V MFlavor.hasRAM] : int 2 TrafficSimulationUF.CRMUnitModel.GigaBytes

attribute CORES

[MetaDataModel.MELODICMetadataSchema.ApplicationPlacementModel.laaS.Processi ng.CPU.hasMinNumberofCores] : int 5

location requirement PL [Locations.PL] slo AvailSLO constraint ConstrModel.AvailConstr

metric type model MetrModel{ composite metric context AvgAvailContext{ metric MetricTemplateCamelModel.MetricTemplateModel.AverageAvailability grouping global

```
constraint model ConstrModel{
```

metric constraint AvailConstr : [MetrModel.AvgAvailContext] >= 0.99999

Cloud Application Modeling with the CAMEL Designer

- The MORPHEMIC CAMEL Designer tool, Cloud Application Modelling and Execution Language (CAMEL) is avaiable for Modelio 4.1 on Modelio R&D GitHub at https://github.com/Modelio-R-D/CamelDesigner
- Modelio Open-source Community and environment are available at modelio.org







Q

Thanks!!



www.softeam.fr

Follow MORPHEMIC project @_morphemic_

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 the Softeam Software BU team for their support.

https://www.morphemic.cloud/