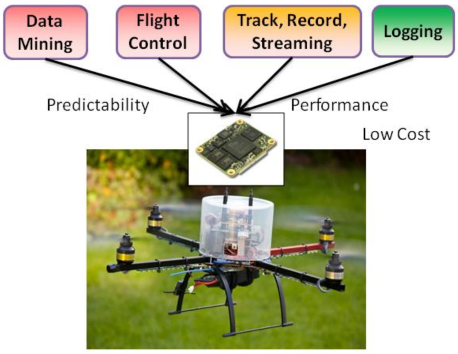
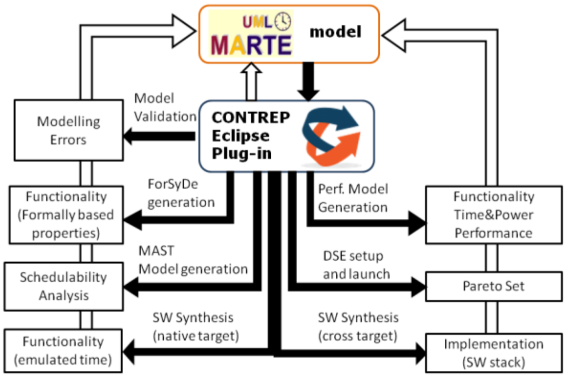
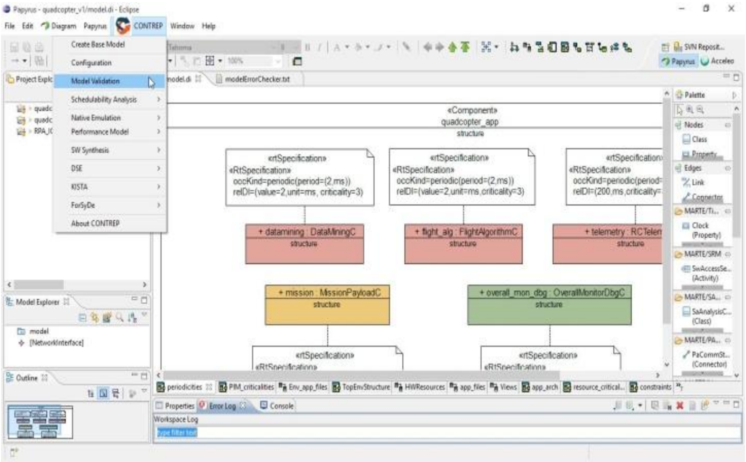
CONTREP: A single-source framework for UML-based Modelling and Design of Mixed-Criticality Systems



Mixed criticality systems integrate applications, platform resources and requirements with different criticality on top of cost-effective, efficient and powerful platforms. A criticality reflects the impact of either a failure of a component or a violation of a requirement, which can range from irrelevant to catastrophic effects. The aforementioned competitive advantages enabled by mixed-criticality systems are pushing its proliferation versus conventional implementation approaches. However, mixed-criticality systems also bring novel, non-straightforward modeling and design challenges. The modeling methodology shall provide means to capture criticality, to associate it to different types of modeling elements, and map it to different safety standards and design contexts. The design methodology shall be able to exploit criticality information from the early design phases, starting from the own system modeling, and until the implementation phase, in order to achieve the most efficient and less costly implementation solution, while every performance requirement stated is fulfilled according the associated level of criticality.


UML MARTE model
Model Validation
Modelling Errors
CONTREP Eclipse Plug-in
ForSyDe generation
Functionality (Formally based properties)
MAST Model generation
Schedulability Analysis
SW Synthesis (native target)
Functionality (emulated time)
Perf. Model Generation
Functionality Time&Power Performance
DSE setup and launch
Pareto Set
SW Synthesis (cross target)
Implementation (SW stack)

This booth presents CONTREX Eclipse plug-in (CONTREP), a framework which supports UML/MARTE based modeling, analysis and design of mixed-criticality embedded systems. CONTREP provides tools and a user front-end for different design activities, which in turn rely on back-end tools, like VIPPE, a fast performance estimation tool; MAST, a schedulability analysis tool; or MOST, for design space exploration. The booth shows a model of a quadcopter control system which integrates safety critical (e.g., flight control), mission-critical (e.g., a video processing payload), and non-critical (e.g., monitoring) functions. The booth shows how mixed-criticality is captured, together with the description of the functional architecture and of the multi-core embedded platform where the system is implemented, and how the framework automates different design activities, i.e. model validation, performance assessment and design space exploration, exploiting mixed-criticality information in every case.


Papyrus - quadcopter_v1/model.di - Eclipse
File
Edit
Diagram
Papyrus
CONTREP
Window
Help
Create Base Model
Configuration
Model Validation
Schedulability Analysis
Native Emulation
Performance Model
SW Synthesis
DSE
KISTA
ForSyDe
About CONTREP
Project Explc
quadc
quadc
RPA_IC
Model Explorer
model
[NetworkInterface]
Outline
model.di
modelErrorChecker.txt
«Component»
quadcopter_app
structure
«rtSpecification»
«RtSpecification»
occKind=periodic(period=(2,ms))
relDl=(value=2,unit=ms, criticality=3)
«rtSpecification»
«RtSpecification»
occKind=periodic(period=(2,ms))
relDl=(value=2,unit=ms,criticality=3)
«rtSpecification»
«RtSpecification»
occKind=periodic(period=
relDl=(200,ms,criticality=
+ datamining : DataMiningC
structure
+ flight_alg : FlightAlgorithmC
structure
+ telemetry : RCTelen
structure
+ mission : MissionPayloadC
structure
+ overall_mon_dbg : OverallMonitorDbgC
structure
«rtSpecification»
«RtSpecification»
«rtSpecification»
«RtSpecification»
SVN Reposit...
Papyrus
Acceleo
Palette
Nodes
Class
Property
Edges
Link
Connector
MARTE/Ti...
Clock (Property)
MARTE/SRM
SwAccessSe... (Activity)
MARTE/SA...
SaAnalysisC... (Class)
MARTE/PA...
PaCommSt... (Connector)
periodicities
PIM_criticalities
Env_app_files
TopEnvStructure
HWResources
app_files
Views
app_arch
resource_critical...
constraints
Properties
Error Log
Console
Workspace Log
type filter text