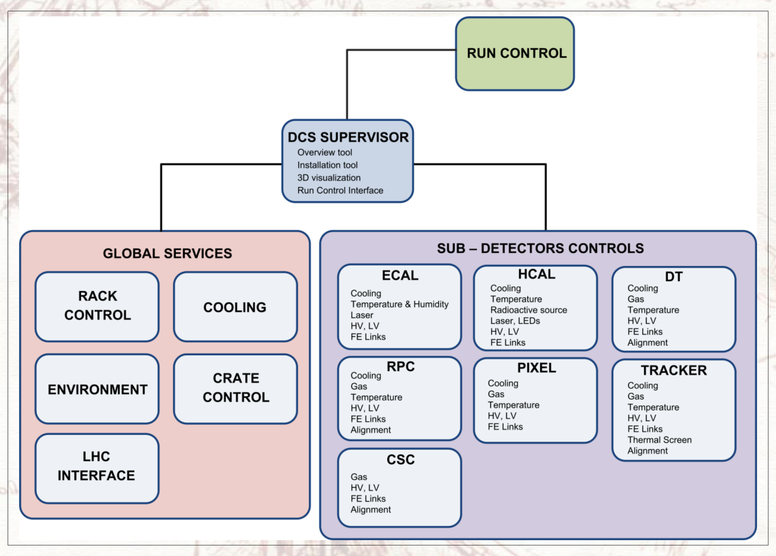
An Analysis of the Control Hierarchy Modelling of the CMS Detector Control System

Y.L.Hwong*, A.Racz, B.Beccati, C.Deldicque, C.Schwick, D.Gigi, E.Cano, E.Meschi, F.Glege, F.Meijers, H.Sakulin, J.A.Coarasa, J.F.Laurens, J.Gutleber, L.Orsini, M.Ciganek, M.Simon, M.Zanetti, R.Gomez-Reino, R.Moser, S.Cittolin (CERN) A.Meyer, D.Hatton, U.Behrens (DESY) D.Shpakov, H.Cheung, J.A.Lopez-Perez, K.Biery, R.K.Mommsen, V.O'Dell (Fermilab) A.S.Yoon, C.Loizides, C.Paus, F.Ma, G.Bauer, J.F.Serrano Margaleff, K.Sumorok (MIT) S.Erhan (UCLA) A.Petrucci, J.Branson, M.Pieri, M.Sani (UCSD) J.F.Groote,T.Willemse (TUE)

Introduction

The high level Detector Control System (DCS) of the CMS experiment is modelled using Finite State Machines (FSM), which cover the control application behaviours of all the sub-detectors and support services.



The FSM tree of the whole CMS experiment consists of tens of thousands of nodes, which makes the implementation of a homogenous and consistent system a non-trivial matter.

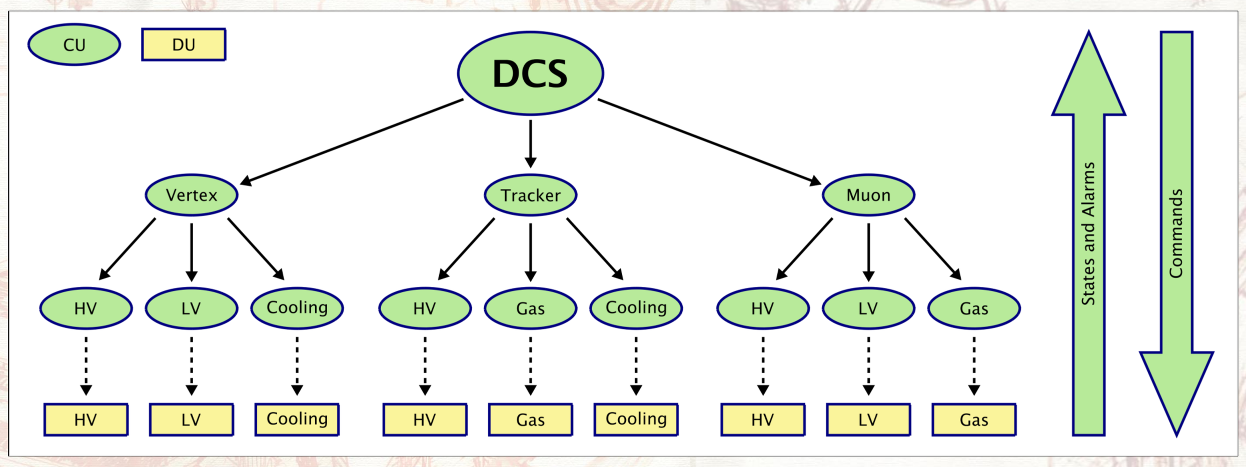
The micro Common Representation Language 2 (mCRL2) analysis technique is being adopted to describe and analyze the CMS FSM system. Using its accompanying toolset, the FSM system of the RPC sub-detector has been analyzed and verified.

Background

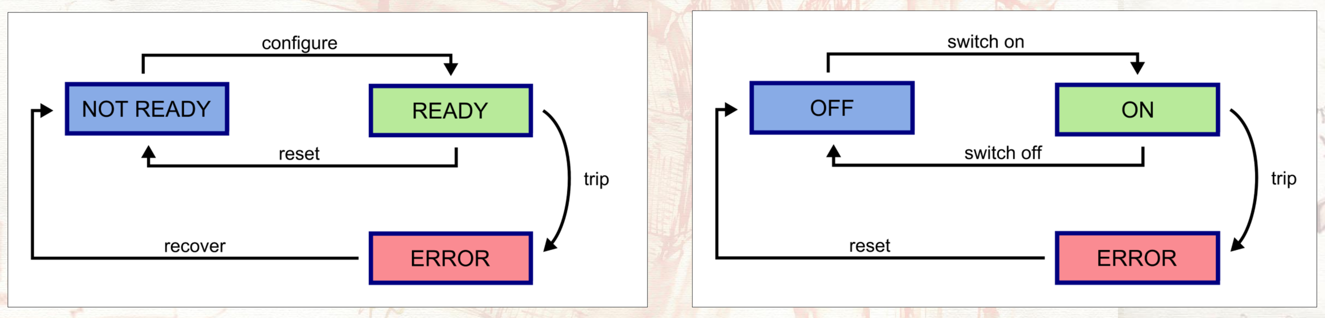
In the CMS FSM model there is a state/command interface between a parent and its children. Commands are passed from a parent to its children and the states of the children are propagated to the parent. Two types of objects are defined in this hierarchy:

- Control Units (CUs): They are the logical units
- Device Units (DUs): They represent a piece of hardware

Each sub-detector group of the CMS experiment is responsible for the development of its own FSM tree control layer modelling their system. CMS will then integrate all these sub-trees into a single FSM tree.



The diversity in the development philosophy of different sub-detector groups and the enormous amount of parameters to be monitored is a fundamental aspect of a large-scale experiment such as the CMS experiment.

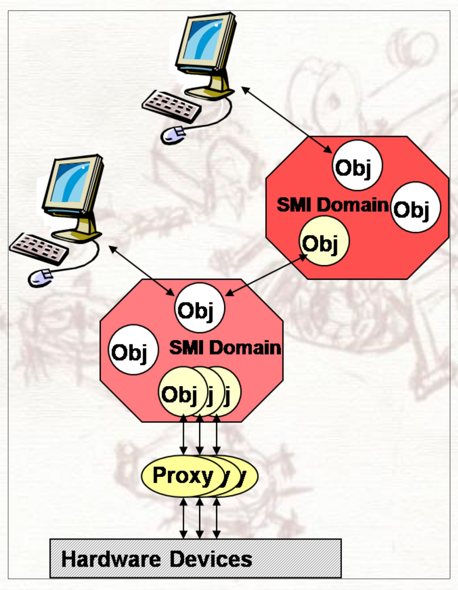


State transition diagram for a Control Unit (left) and a Device Unit (right).

Detector	Monitored Parameters	Channels
Muon Systems	HV,LV,FEE and others	124.2 k
Calorimeters	HV,LV,FEE and others	110.9 k
Trackers	HV,LV,DCU and others	27.7 k
	Total	262.8 k

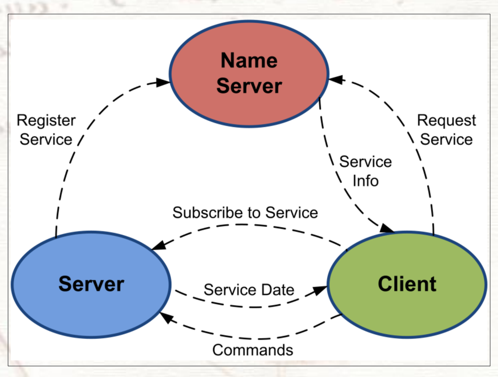
Parameter overview of the CMS DCS

Architecture



The Joint COntrol Projects (JCOP) at CERN has chosen the SMI++ framework for the modelling of the FSM systems. Logically related objects are grouped into SMI++ domains; in which the objects are organized in a hierarchical structure and form a sub-system control.

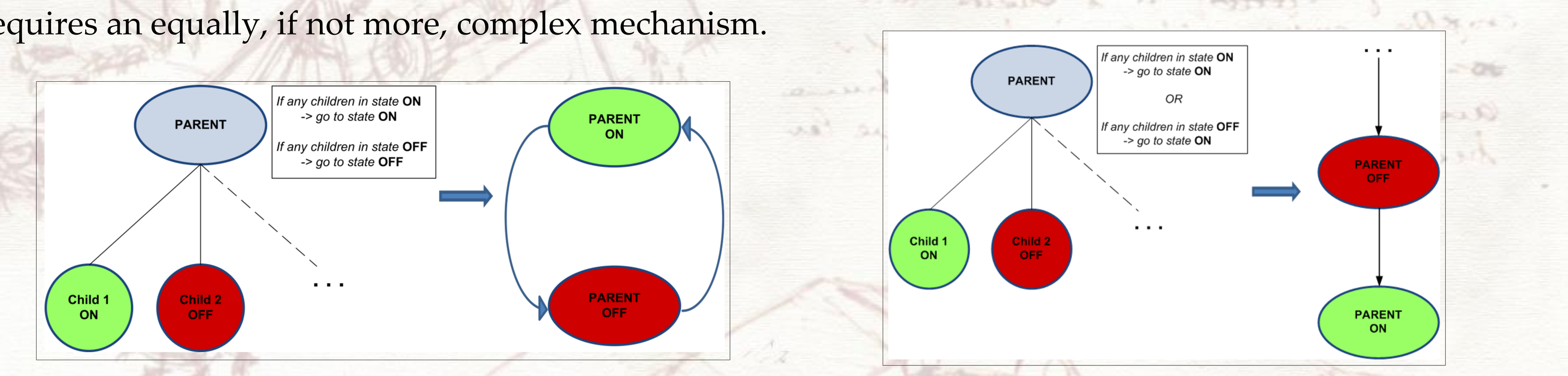
The distribution and data exchange of this large system is handled by DIM (Distributed Information Management system), which is based on the client / server paradigm.



Objective

To ensure a sound logic implementation throughout a system of such size, the investigation of some desired properties such as deadlock and endless-loop freedom requires an equally, if not more, complex mechanism.

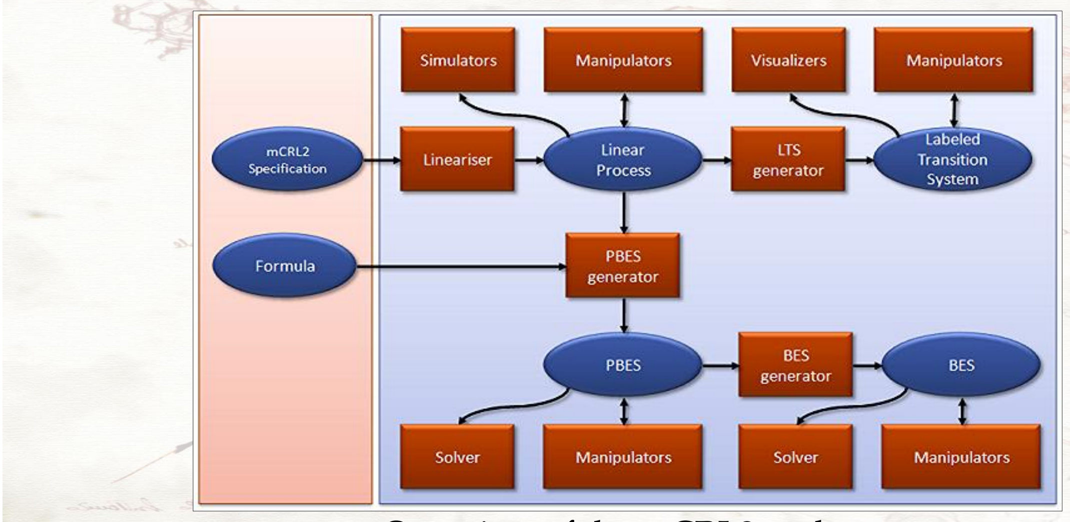
The modelling and analysis of the CMS FSM system is a challenging task. The application of the mCRL2 toolset for this purpose is investigated.



An illustration of how an endless loop (left) and a deadlock (right) could happen.

Philosophy

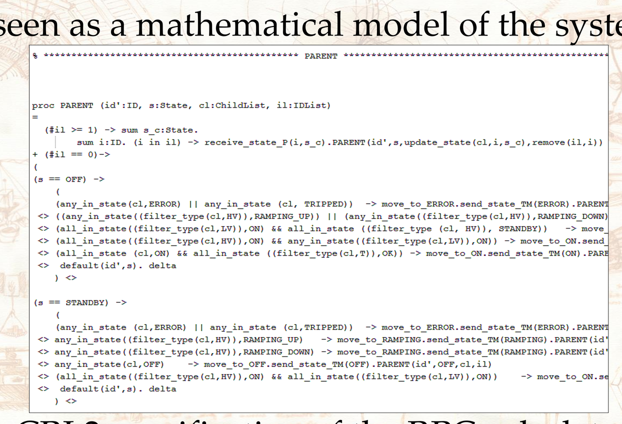
mCRL2 is a specification language that can be used to specify and analyze the behaviour of large distributed systems and protocols. The language is supported by a toolset enabling simulation, visualisation, behavioural reduction and verification of software requirements.



Overview of the mCRL2 toolset

Specification

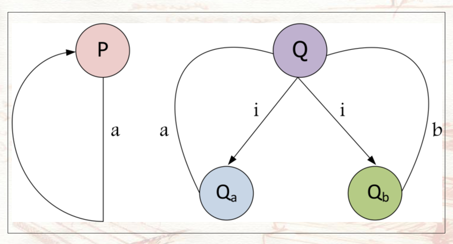
A mCRL2 specification is a plain-text file containing a model in the mCRL2 language. It can be seen as a mathematical model of the system.



mCRL2 specification of the RPC sub-detector.

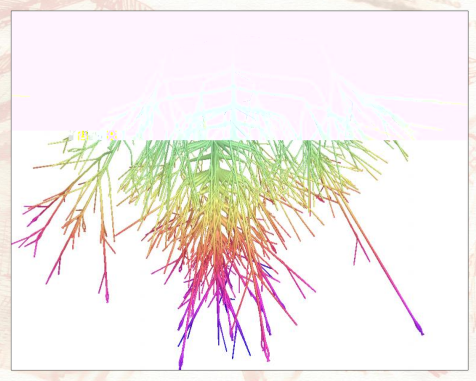
Linearization

The first step in the mCRL2 analysis process is to linearize the specification to obtain a *Linear Process Specification* (LPS). All that remains in a LPS is a series of condition – action – effect rules that specify how the system as a whole reacts to a certain stimuli.



$$P \,//\, Q \rightarrow a.(P \,//\, Q) + i.(P \,//\, Q_a) + i.(P \,//\, Q_b) + a|i.(P \,//\, Q_a) + a|i.(P \,//\, Q_b)$$

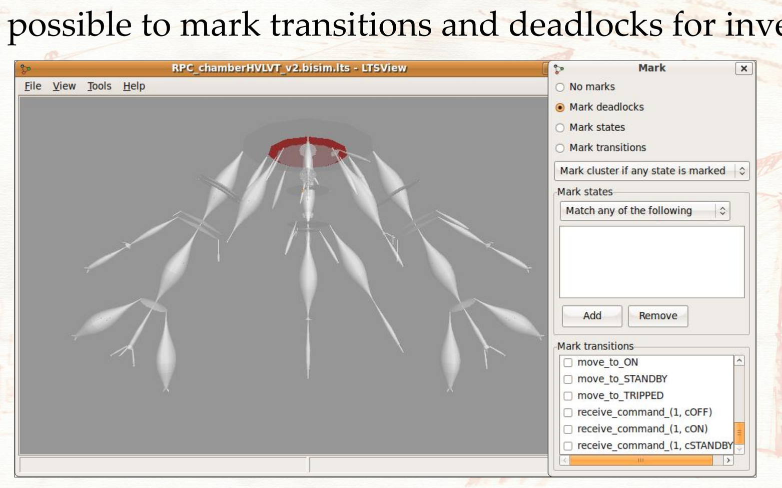
LPSs are a compact symbolic representation of the state space of the specification. Most mCRL2 toolset operate on LPS rather than on state spaces.



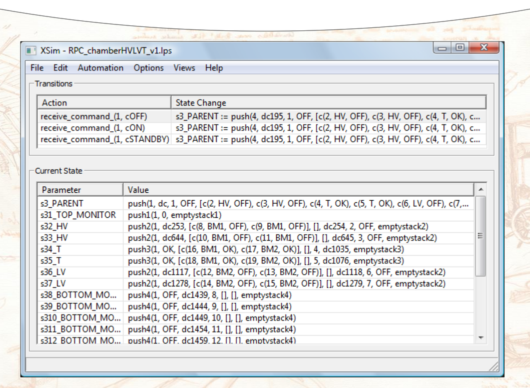
mCRL2
micro Common Representation Language 2

Visualization

From the LPS, a *Labelled Transition System* (LTS) can be generated. A LTS is an explicit representation of the state space and can be visualized using interactive GUI tools. A sophisticated way of visualization is through 3D representation by employing a clustering technique to reduce the complexity of the image. The visualizations help in scrutinizing the model based on unexpected visual anomalies. It is possible to mark transitions and deadlocks for investigation.



Simulation

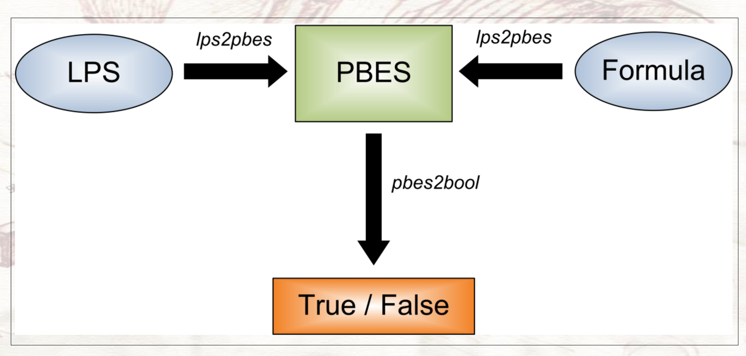


From the initial state, sequences of actions can be performed to reveal unexpected or erroneous behaviour. It is possible to manually trigger state changes, or traces can be generated automatically.

Model Checking

Model checking is a verification method to show that the system exhibits certain desired properties. Given a LPS and a formula (expressed in the *regular modal µ-calculus*), a *Parameterized Boolean Equation System* (PBES) is generated.

The model checking question of "does the formula hold for this LPS" is encoded in the PBES. By solving this PBES, an answer (true/false) to this question can be found.



Conclusions

The mCRL2 is a versatile and powerful toolset for the study and modelling of large distributed system, as is seen by its adoption for the analysis and optimization of the CMS RPC FSM system. Further application on other parts of the system is planned and is envisaged to enhance the performance and provide a better insight into the whole CMS FSM system. However, proper training is required for an accurate modelling of the system in such a way that its properties can be efficiently checked using the toolset. As the toolset harbours great potential, an automated tool for the translation from the FSM to the mCRL2 language for verification purposes is foreseen to be a highly rewarding project for the future.

Acknowledgements

I would like to express my gratitude to Jan Friso Groote and Tim Willemse from the Design and Analysis of System group at the University of Technology Eindhoven for their generous help and support in the course of this research project.

* Marie-Curie Early Stage Researcher: yi.ling.hwong@cern.ch

This research project has been supported by a Marie Curie Initial Training Network Fellowship of the European Community's Seventh Framework Programme under contract number (PITN-GA-2008-211801-ACEOLE)

ACEOLE www.cern.ch/aceole

Technische Universiteit Eindhoven University of Technology

This research project is a collaboration with the department of Computer Science of the Eindhoven University of Technology.