Dr Tim Willemse and Professor Dr Jan Friso Groote discuss their involvement in an international team of researchers who have developed specific software to detect inherent problems in the control system of the Compact Muon Solenoid experiment at CERN.

To provide some background, can you briefly describe the Compact Muon Solenoid (CMS) experiment at CERN’s Large Hadron Collider (LHC)? How is software controlling the experiment structured?

TW: The CMS experiment at CERN is one of four large physics experiments designed to study the wide range of particles produced from the high energy collisions in the LHC. The software controlling this experiment has a very hierarchical structure and consists of 25,000-30,000 components, each behaving as finite state machines (FSMs). These are arranged in a tree structure and only communicate with FSMs that are immediately above or below them. Control commands are propagated and refined up the tree of FSMs. State changes of FSMs are sent down the tree. This architecture allows the immense control problem to be decomposed into smaller, more manageable problems.

Are there concerns with CMS’s hierarchical control software for which you have devoted your research efforts in this project?

TW: CERN’s engineers observed that subsystems of the control system became non-responsive from time to time. Control commands were being sent but the control system was not reacting to them. This non-responsiveness meant that there was no easy way to prove this. When asked if we could help identify the root cause, we started by modelling parts of the control system and applying our tools to analyse these parts. This allowed us to unearth several issues that caused the control system to become non-responsive. Based on these findings, we have developed dedicated tools that are now incorporated in the development process of CMS’s FSMs.

What is your specialism and who comprises your team?

TW: We specialise in analysing complex software systems. Our background enabled us to quickly zoom in on the core of the problem in the control system. For instance, we had reason to believe that the non-responsiveness of the control software was caused by subtle software bugs that cause a subsystem to enter into an endless computation, flooding the network. These bugs – called livelocks – are virtually impossible to find using testing. Our technology allows such bugs to be detected and resolved. Indeed, for the control system of the CMS experiment, we detected a large number of different livelocks.

Our project team consists of computer scientists from Eindhoven University of Technology and Twente University in The Netherlands, Dr Frank Glege, a physicist at CERN, and Professor Dr Rance Cleaveland from the University of Maryland, USA. Most of the theoretical research is carried out by two PhD students: Maciej Gazda, working at Eindhoven University of Technology, supervised by Groote and me; and Gijs Kant, working at Twente University, supervised by Professor Jaco van de Pol. We have also worked with excellent Master’s students from our university and from CERN, and also one of our PhD students, Jeroen Keiren.

Do you foresee similar problems cropping up in the other experiments at the LHC, and also in experiments at other facilities?

TW: As it turns out, the other large experiments running at the LHC use the same architecture for controlling their experiments. The technology that we have developed for CMS is now also used for the other experiments.

JF: The phenomena that we’ve observed in CMS’s control system are really universal to all companies that produce systems involving large-scale software solutions. For example, we observed similar phenomena in the Atacama Large Millimeter/submillimeter Array (ALMA) project run by the European Southern Observatory.

How will research progress over the coming months and years? Will you be changing your research focus?

TW: At the beginning of the project, we mainly focused on solving problems our collaborators at CERN found important. Using our existing theory and tools, we were indeed able to pinpoint the causes of those problems, and solve them. But some of the problems that were deemed to be less important at that time actually posed more of a challenge to us. We are now working on improving our theory to be able to effectively solve those problems as well. We are anxious to apply our technology to other systems within CERN and elsewhere.
Under control?

The increasing complexity of control systems used in experiments at the Large Hadron Collider has made it necessary to develop toolkits to model and analyse them in order to detect and fix the problems that arise.

THE COMPACT MUON SOLENOID (CMS)

The experiment at the Large Hadron Collider (LHC) near Geneva is a particle detector with a diameter of 15 metres and a weight of approximately 15 tonnes. It consists of seven subdetectors which are capable of stopping, tracking and measuring the particles produced by high energy proton collisions. The LHC accelerates protons to a velocity close to that of light in order to create head-on collisions, and the energy released from this process is transformed into mass in the form of short-lived particles. CMS serves to observe and study the particles and phenomena resulting from these high energy reactions. Whilst it makes more accurate measurement of the properties of known particles, it is also on the lookout for novel phenomena.

CONTROL SYSTEM

The experiment is monitored in real time by over 27,500 finite states machines (FSMs) organised in a hierarchical manner to form the control system. These FSMs are relatively simple, with each one having an average of five logical states, each state being in one of two possible phases. However, complexity arises out of the interaction between the components of the control system. At the top of the hierarchy lies a single controlling FSM, whilst the bottom typically consists of myriad devices (both hardware and software). Commands are sent downwards whilst status and alarms travel in the opposite direction. The average depth of the architecture is nine nodes, with a maximum depth of 11 and a minimum depth of three. The principal role of this hierarchical control system is to switch the detector on and off.

Unfortunately, this complexity inherent to the system is such that recently the entire system became unpredictable. The software lost track of parts of the experiments and, whilst the CMS researchers could analyse clusters of FSMs, the problems faced by the team required complete system verification. Because the CMS research group comes largely from a physics background, the researchers called on the expertise of Dr Tim Willemse from Eindhoven University of Technology, who leads an international team of scientists.

"For systems of this size and complexity, a thorough understanding of the problems can only be obtained by modelling the essential parts of the system and analysing these models," he explains.

FINDING A LANGUAGE

In order to analyse and model the control system, it is first necessary to translate the hierarchy of FSMs (described in State Manager Language) into a language suited to the problem at hand. There exist many logics of different orders with varying expressive powers and more or less complex syntaxes. To carry out the task, the team needed to find a sufficiently expressive logic which was not unnecessarily complicated. "Propositional logic reasons about propositions that can be either true or false, and their conjunction, disjunction and logical negation, but it has no variables over which one can quantify," Willemse expands. "In a nutshell, first-order logic essentially extends propositional logic by adding variables over which one can quantify."

The logic chosen by the team is an extension of this first-order logic called parameterised mCRL2 could model, construct and solve parameterised Boolean equation systems, thereby providing a suitable way of analysing the complex architecture of the CMS control system as well as other systems with similar architectures.
The phenomena found in the CMS control system are shared by other software including complex systems produced by private companies. In addition, companies in The Netherlands such as Verum and Imtech which develop complex embedded software use academic toolsets such as mCRL2 for verification purposes. This has opened the door for multinationals like Philips to propose using formal methods in the development of their products.