Model-driven quality and resource management for CPSs

Model-driven Quality and Resource Management for CPS

Freek van den Berg[1] Marc Geilen[1] Twan Basten[1]

Keywords: Quality and Resource Management, Model-Driven Design, Domain-Specific Languages, Component-Based Design, Cyber-Physical Systems.

Motivation: A Cyber-Physical System (CPS) integrates cyber systems, human users, networks and physical systems. Consequently, a CPS needs visual context and awareness to make autonomous and correct decisions. Advanced image and video processing is computationally intensive and challenging. Moreover, a CPS comprises increasingly complex and distributed configurations, which is reflected in the growing number of sensors, actuators and other smart devices. This leads to an exponential number of dynamic system configurations. To make matters worse, a CPS needs to simultaneously satisfy many rigorous constraints, e.g., hard deadlines, safety, quality, and performance. Hence, the system designer is confronted with an immense number of potential configurations of which only a limited number meet the constraints and only a fraction are optimal regarding certain qualities. This makes finding the optimal configurations hard, especially during run-time. A domain-specific language (DSL) for quality and resource managment (QRM) is presented to specify these configurations conveniently and reason about them in an automated manner.

Case study: We investigate aspects of the interventional X-Ray (iXR) systems of our project partner Philips Healthcare. An iXR system (see Figure 1) assists a surgeon during an operation by providing a continuous stream of images of the inside of a patient's body on a display. These images are based on X-ray beams and require intensive image processing to be maximally helpful for the surgeon. In this case study, we focus on the situation in which a dynamically varying number of image streams of varying qualities are shown on the display. The qualities of concern are the end-to-end image latency with a hard constraint and frame-rate of the individual image streams, whereas the hardware resources are limited and fixed.

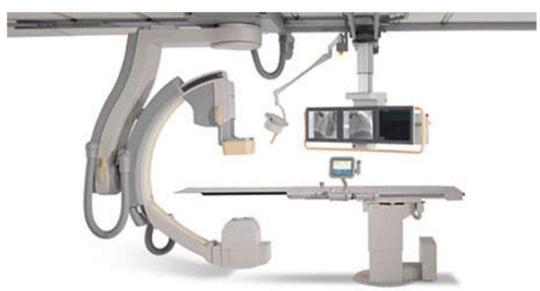

Fig. 1: an iXR machine

[1] Electronic Systems, Eindhoven University of Technology, Netherlands

Method: We propose Quality and Resource Management Language [2,3] (QRML) and its corresponding toolset that allow a system designer to specify and analyze configurations, constraints and objectives in a component-based way.

A QRML model consists of a hierarchy of components. A component: (i) receives input; (ii) provides output; (iii) provides budget; (iv) requires budget; (v) exhibits configuration parameters; and, (vi) exhibits qualities to be used to express constraints and requirements for quality and resource optimization purposes. A component is either an atomic or composite component. A composite component is either a component aggregation, i.e., a set of collaborating components, or a set of component alternatives of which one is selected.

The iXR QRML model. The modeled iXR image processing system comprises a number of applications, one so-called virtual execution platform (VEP) for each application, and a physical platform. An application processes a stream of images with qualities frame-rate and latency. The platform is fixed and includes network components, memories and video scalers that adapt image sizes. An application requires budgets captured by its VEP. That is, a VEP virtualizes platform resources for an application. Each VEP binds one application to the platform by providing budgets to an application and requiring budgets from the platform. Typical budgets are network bandwidths, processing cycles and memory. The bindings of the VEPs to the platform can be realized in several ways. This gives rise to many potential configurations of which only a few meet the platform and application constraints. Based on the qualities, the optimal configurations are the ones that perform best on the frame-rate and latency of the different applications. The number of image streams and their qualities may vary during run-time, which calls for a solution that can be applied at run-time.

Automated design support for QRML. QRML is a versatile toolset which provides automated support to the system designer. First, the system designer can take advantage of navigation, syntax highlighting and input completion while creating a model, which is based on the underlying grammar of QRML. Second, the system designer receives continuous input validation while creating a model. Third, a QRML model is visualized in various ways for increased human comprehension and communication. Fourth, the potential, feasible and optimal configurations are derived from a QRML model. As a future addition, a transformation from QRML to executable software that dynamically selects configurations upon system changes will be developed.

Results: QRML was used to model aspects of an iXR system. During model creation, QRML provides syntax highlighting, input completion and validation. QRML creates multiple visualizations that increase understanding of the system. Finally, QRML derives the expressed constraints that can be used to determine the feasible and optimal system configurations using a constraint solver.

---

[2] qrml.org, the official QRML website.

[3] M. Hendriks, M. Geilen, K. Goossens, R. de Jong, T. Basten. Interface Modeling for Quality and Resource Management, arXiv:2002.08181, 2020. Background.