Poster: Towards a Secure, Resilient, and Distributed Infrastructure for Hydropower Plant Unit Control

Andrea Höller, Johannes Iber, Tobias Rauter, and Christian Kreiner
Graz University of Technology
{firstname.lastname}@tugraz.at

Abstract

Today, there are ever increasing demands on hydroelectrical power plant controllers. They have to deal with a power grid that becomes ever more unpredictable due to renewable energies. Furthermore, power plants represent a critical infrastructure that have to provide a full operation, even in the presence of cyber-attacks and internal faults.

Here, we present an approach towards tackling these challenges by integrating the knowledge of multiple research domains such as security, fault tolerance and modeling. We propose a distributed infrastructure that provides resilience via an assured dynamic self-adaption. Further, we show the integration into an existing hydropower plant unit control.

1 Introduction

For nearly a century the electrical protection, the generator voltage regulation and synchronization with the power grid of hydropower plant units was done by specialized mechanical and electromechanical devices. This is currently changing, because other renewable energy sources like wind or solar are integrated into the power grid on a grand scale [4]. Their complicated predictability concerning energy conversion has an impact on the technology of hydropower plant unit control systems because nowadays these have to react on power grid changes in time to achieve overall grid stability. At the same time, these power plants represent critical infrastructures that have to be protected against cyber-security attacks. Furthermore, a shutdown of the plant due to faults in the control device can lead to economic losses, and even large scale blackouts.

To tackle this challenge, we perform research together with our industrial partner on how to create future resilient and secure power plant controllers. As far as the authors of this work know, self-adaptability has not been applied in this domain so far. We combine research from different fields like security, fault tolerance, and models@run.time to enhance the resilience of an existing control system. These proposed principles are also applicable for cyber-physical systems in other domains.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2016
15–17 February, Graz, Austria

ISBN: 978-0-9949886-0-7

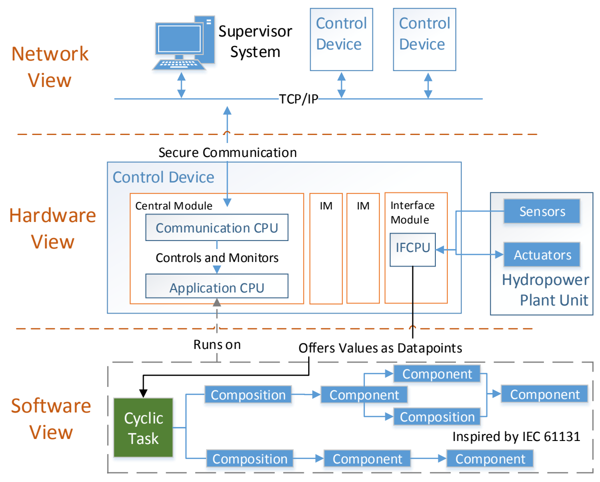

Figure 1. Overview of the existing system

2 Existing Hydropower Plant Control System

Figure 1 illustrates the existing target system of our proposed infrastructure. On network level the control devices are operated by a so-called supervisor system and connected by wire. The supervisor system is mainly responsible for deploying tasks on devices and collecting data from the single hydropower plant control devices.

The control devices are connected to hydropower plant units. Their functional responsibility is to operate hydropower plant units through the different phases excitation, synchronization, protection and turbine control.

Technically, these control devices have a programmable logic controller (PLC) architecture. Concerning the hardware, a control device is build out of central modules and interface modules. A central module consists of a communication CPU and an application CPU. The communication CPU is responsible for network connections and controlling/monitoring the application CPU, and runs a customized Linux. From the security point of view it also acts as gateway for the application CPU. The application CPU is a multi-core processor and executes the actual logic. It runs a real-time operating system in order to ensure guaranteed cycle times. The interface modules are connecting the control device with sensors and actuators of the hydropower plant unit.

The software executed by the application CPU of a central module is component-based and inspired by the IEC 61131 standard for programmable controllers. Basically, the software is hierarchically build out of components, compositions and tasks. Components are coded with the C-programming language and stored as binaries on the devices. Such components implement basic function blocks, e.g. simple logic gates, or complex control algorithms. Based on these components, compositions are designed that implement the specific control logic for a hydropower plant unit. Finally, such compositions are called by cyclic tasks.

The compositions operate on so-called datapoints that are set and read by the interface modules. At the start of a cyclic task the necessary datapoints are collected, the compositions are executed, and subsequently the calculated datapoints are written back.

From the plant engineer's point of view this system is configured with model-driven techniques. Domain-specific modeling languages (DSL) are used to describe tasks, compositions, and components. As well, resources consisting of modules, control devices, networks, and connected hydropower plant units are modeled. Finally, deployment of tasks onto control devices is specified by yet another DSL. Concerning extra-functional properties (e.g. timing, security, memory consumption, . . .), we rely on contract-based design [3]. All these models are traditionally used at design time. Additionally, we are going to leverage them at runtime for the infrastructure proposed.

3 Proposed Infrastructure

We enhance the control device with security and hardware fault monitoring. Observed anomalies are forwarded to the supervisor running a reasoner application that interprets these anomalies and executes mitigation strategies.

3.1 Control Device

The control device has to fulfill high performance requirements, since a lot of sensor data has to be processed in short time. However, the nature of the control application is well-suited for tuning the performance with parallelization. Considering dependability, the use of highly integrated multi-core is a double-edged sword: future semiconductor technologies are expected to be very susceptible to hardware errors due to small structures, however additional processing units also allow the integration of additional fault tolerance techniques.

More precisely, we exploit the unused computing capacity to enhance the diagnostic features regarding hardware faults with spatial redundancy. Cores that are not needed to realize the main functionality are used to perform partial redundant calculations, cross-checking the achieved results, and analyzing the trend of observed errors. This error trend is used to distinguish between transient and permanent faults.

In order to increase the diagnostic capabilities, the redundant calculations are done in a diverse way. For introducing the diversity in the redundant replicas, we use cost-efficient ways to automatically introduce diversity in execution as proposed in [2]. For example, different compiler and compiler flags can be used to generate several binaries that perform the same calculations, but show different execution characteristics (e.g. timing, hardware resource usage). We have shown that this approach is well-suited to detect common-cause faults such as memory-related software bugs, or hardware faults in shared resources [1].

Furthermore, we enhance the resilience regarding malicious security attacks. First, the diverse redundancy concept expenses the required effort to attack the functional application, since all redundant executions have to be attacked simultaneously. Anomaly-detection is performed on the communication CPU. This includes checking the plausibility of sensor data, network anomalies, and the integrity of other devices in the network [5].

3.2 Reasoner

The reasoner manages a model of the system that includes functional and extra-functional properties of the target application [3]. Furthermore, it receives information about detected anomalies from the control device. By analyzing these data, the reasoner decides, whether and how to reconfigure the system. For example, if hardware or software faults affect the control device a fault recovery strategy is executed. This fault recovery is realized by updating the component binaries with diverse binary versions [2]. The system model is used to assure that the extra-functional requirements (e.g. timing/memory constraints) are still fulfilled after the update.

If these strategies are not successful, the system is reconfigured in such a way that the faulty device is isolated or the control logic is redistributed. Again, the system model is used to assure a correct reconfiguration. Furthermore, the reasoner includes a detection mechanism that distinguishes between non-malicious faults (e.g. software- or hardware faults) and malicious faults. If it is possible to identify the attacked part of the system, this part is isolated. Every reconfiguration alarms the power plant operator.

4 Conclusions

The primary aim of our work is to enhance the security and resilience of hydropower plant unit controls. We propose a distributed infrastructure which reconfigures itself based on the observed hardware faults or security violations. This mechanism is going to leverage models at runtime which are a product of the design-time configuration.

5 References

[1] A. Höller, N. Kajtazovic, K. Römer, and C. Kreiner. Evaluation of Diverse Compiling for Software Fault Tolerance. In *DATE*, 2015.

[2] A. Höller, T. Rauter, J. Iber, and C. Kreiner. Software-Based Fault Recovery via Adaptive Diversity for COTS Multi-Core Processors. In *ADAPT Workshop*, 2016.

[3] J. Iber, A. Höller, T. Rauter, and C. Kreiner. Towards a Generic Modeling Language for Contract-Based Design. In *ModComp Workshop (MoDELS)*, 2015.

[4] M. Liserre, T. Sauter, and J. Hung. Future Energy Systems: Integrating Renewable Energy Sources into the Smart Power Grid Through Industrial Electronics. *IEEE Industrial Electronics Magazine*, 4, 2010.

[5] T. Rauter, A. Höller, J. Iber, and C. Kreiner. Thingtegrity: A Scalable Trusted Computing Architecture for Resource Constrained Devices. In *EWSN*, 2016. to appear.