Deploying Constraint Programming for Testing ABB's Painting Robots

Morten Mossige, Arnaud Gotlieb, Hein Meling

■ *This report explores the use of constraint programming for the validation of ABB Robotics' painting robots.*

Testing software systems is a difficult cognitive task that requires human skills to generate test scenarios and to predict the expected behaviors of a system. Moreover, coming up with possible failure scenarios that may arise during system operation and determining triggers for these cases is a major challenge.

ABB Robotics' painting robots are advanced distributed systems that must be thoroughly tested before they are shipped to customers, mostly from the automotive and avionics domains. Improving the overall quality of painting robots has always been of utmost importance because it is viewed as a competitive advantage in the market for industrial robots — an error in the control system of these robots, uncovered in the field, can have serious economic consequences. Among the many possible failure sources are those

ISSN 0738-4602

related to timing the triggering of different subsystems. For example, an overlap-error scenario can occur when the robot's speed increases and switching between different paint spray patterns occurs too rapidly. In principle, the control system should detect overlap scenarios, send an appropriate error message, and possibly shut down the system. However, it may happen that the robot does not work as expected, for example due to bugs in the design or implementation of the control system. Creating test scenarios targeted at finding such subtle faults is nontrivial.

Faced with this challenge, researchers and engineers from Certus, a Norwegian research-based innovation center, have teamed up to explore the usage of constraint programming (CP) for the validation of ABB's painting robots. Constraint-satisfaction techniques over finite domains were selected because of their versatility in dealing with heterogeneous constraints derived from the design and implementation of ABB's control system for paint robots. We used a combination of constraint propagation with different filtering consistencies and dedicated search heuristics. With these techniques it was possible to propose a cost-effective solution for test case generation for validating the control systems of ABB's painting robots (Mossige, Gotlieb, and Meling 2015; Mossige, Gotliev, and Meling 2014a, 2014b). Currently, these AI techniques are commonly supported by modern CP solvers that ease their adoption in industrial contexts. However, using a constraint solver to generate tests can be time consuming, which is at odds with the need to run both test generation and test execution as part of a continuous integration cycle. Indeed, continuous integration is a software engineering practice, where the result of test execution, following a source code change, should be reported back to the developer quickly.

Despite the strong technical culture of ABB Robotics and the availability of excellent textbooks, deploying CP was not easy, as there are almost no guidelines on how to model with constraints and how to integrate CP in an industrial software production process. However, we managed to develop such a model and to deploy it at ABB Robotics by adding the constraint-solving process to the continuous integration process. The model has now been part of ABB's continuous integration process for more than 24 months of daily operations. After having collected data about its bug-finding capabilities and efficiency, we perform a thorough analysis of its benefits and drawbacks, which led us to draw some lessons learned from this experience.

Bug-Finding Capabilities of ABB's CP Model

For the purpose of validating the CP model itself, known bugs were reintroduced in the paint control systems to examine the capabilities of the model to find them. After some tuning, the model was able to generate test scenarios that could find all the re-introduced bugs. This was considered a significant success and justified the continuation of the research and innovation activities. Next, immediately after the integration of the model in the software production system, several unknown, but subtle bugs were found related to the timing aspects of the paint control systems. However, these bugs were classified as noncritical, as they represented very unlikely scenarios. Upon further analysis, we observed that these bugs had been present in the control systems for several years without any significant consequences. They were corrected and the generated test scenarios were added to the testing process. However, although the bug-finding capabilities of the model had been strongly validated, ABB believed that its ability to uncover mainstream bugs still remained to be demonstrated.

Deploying CP in Industrial Software Validation

The validation of paint robots involves a fair amount of manual work, which is also error prone by nature. Therefore, deploying CP as a way to automate some parts of the process was welcomed by validation engineers and perceived as a way to strengthen the validation process. However, CP also comes with some challenges. The adoption of CP requires training, and the maintenance of the CP model without expert assistance is difficult, especially since the automatically generated test scenarios are difficult to understand for untrained engineers. Tuning the model and its solving parameters, especially when optimization procedures are used, turned out to be reserved for experts only. Validation engineers are usually skeptical about tools that produce results that are incomprehensible to them, and almost impossible to compute by hand. To reduce the risks of rejection, internal training sessions on CP were organized and various front ends were built to help the engineers manage the complexity of the generated test scenarios.

Return on Investment on Using CP

To compute the return on investment, one can measure the number of uncovered bugs with and without the CP model or compare the human effort required in both cases. However, for us, another more important factor was evaluating the possibly increased confidence of the engineers in the validation process — a factor almost impossible to quantify. After having deployed CP, we observed an increased appetite among engineers to refactor code covered by the CP-generated tests. Such refactoring is needed, but often deferred due to code complexity, which over time can result in technical debt for the system. Upon further analysis, we understood that this came from their increased confidence in the validation process as it acted as a safeguard to detect undesirable side effects.

Improving the overall quality of painting robots has always been of utmost importance, because it is viewed as a competitive advantage. For this study, we chose to automate tests that covers the most critical aspects of the system, as well as tests that were the most laborious to conduct manually.

While the goal of the project was not aimed at cost savings, we did observe that validation engineers are now relieved to focus on other tasks that have yet to be automated. In the long run, however, we expect to detect more bugs before shipping a product, rather than detecting bugs at customer premises, where they are typically much costlier to fix.

The deployment of CP as a part of a continuous integration process at ABB Robotics, Norway, was considered to be a success. It led the company to engage in an ambitious deployment plan at the worldwide level.

References

Mossige, M.; Gotlieb, A.; and Meling, H. 2015. Testing Robot Controllers Using Constraint Programming and Continuous Integration. *Information and Software Technolog,* 57(January): 169–185.

Mossige, M.; Gotlieb, A.; and Meling, H. 2014a. Testing Robotized Paint System Using Constraint Programming: An Industrial Case Study. In *Testing Software and Systems: 26th IFIP WG 6.1 International Conference Proceedings*, Lecture Notes in Computer Science 8763. Berlin: Springer.

Mossige, M.; Gotlieb, A.; and Meling, H. 2014b. Using CP in Automatic Test Generation for ABB Robotics' Paint Control System. In *Principles and Practice of Constraint Programming: 20th International Conference Proceedings,* Lecture Notes in Computer Science 865, 25–41. Berlin: Springer.

Morten Mossige is a principal engineer at ABB Robotics in Bryne, Norway. He is also an adjunct associate professor at the Department of Electrical Engineering and Computer Science in the University of Stavanger, Norway.

Arnaud Gotlieb is leader of the Certus research-based innovation center, Simula Research Laboratory, Fornebu, Norway.

Hein Meling is a professor in the Department of Electrical Engineering and Computer Science at the University of Stavanger, Norway.