Ravensburg Weingarten University

Bachelor Thesis

Developing a Concept for Automated Testing of IPC Middleware – Demonstrated with eCAL

Author: Emircan Tutar

Address: Heyderstr. 7

City: 88131 Lindau

Student ID: 35606

Program: Applied Computer Science

1st Supervisor:

Prof. Dr. Marius Hofmeister
Ravensburg Weingarten University

2nd Supervisor:

M.Sc. Kerstin Keller
Continental

April 11, 2025

Introduction

Inter-process communication (IPC) plays a fundamental role in modern software systems, especially in distributed environments where processes must coordinate reliably across different nodes. IPC middleware solutions abstract the complexities of data exchange, providing a scalable and efficient foundation for inter-process collaboration. However, testing such middleware-based communication reliably—especially under realistic and distributed conditions—remains a significant challenge.

This thesis addresses the question of how a structured and automatable testing concept for IPC middleware can be developed. The study focuses on the middleware framework eCAL (Enhanced Communication Abstraction Layer), which is increasingly used in industrial applications. While eCAL provides tools for publishing, subscribing, and monitoring communication, it lacks a formal, reusable testing strategy—especially for integration-level testing scenarios.

Motivation and State of Research

Industries such as automotive, robotics, and industrial automation increasingly rely on distributed architectures, where correct and reliable inter-process communication is essential. Middleware like eCAL provides a communication backbone, but also introduces a new abstraction layer where integration-level errors may occur, making effective testing more difficult. Traditional unit testing is insufficient in such environments, where timing, message loss, or multi-node synchronization issues can occur.

Other middleware platforms like ROS or DDS offer more mature testing ecosystems, including built-in simulation tools and integration test suites. However, due to architectural and implementation differences, these cannot be directly applied to eCAL. Moreover, academic literature offers limited guidance on testing middleware-specific IPC behaviors such as publish-subscribe latency, message ordering, or fault recovery.

This thesis aims to bridge this gap by (1) analyzing the requirements for testing IPC middleware, (2) designing a modular and extensible test concept tailored for IPC behavior, and (3) evaluating suitable tools for automation. Although eCAL is used as the demonstration platform, the goal is to develop a methodology that is transferable to similar middleware systems.

Research Objective

The aim of this thesis is to create and test a concept for integration testing in IPC middleware. The focus is on:

- Understanding the requirements for integration tests in IPC systems
- Designing a reusable and automatable testing strategy
- Creating example tests using tools like Google Test or Robot Framework
- Showing how the concept works using eCAL as a case study
- Checking if this testing approach can be used in CI/CD pipelines

Research Questions

This thesis answers the following questions about testing IPC middleware:

1. **What are the main requirements for building integration tests in IPC middleware?**
   Focus on aspects like timing, data correctness, and error handling during communication between components.
2. **How can we design an integration test strategy that supports the specific needs of distributed IPC systems?**
   Focus on creating a general concept that works with middleware platforms using test setups.
3. **Which test automation tools are a good fit for integration testing in IPC middleware, using eCAL as an example?**
   Focus on evaluating tools like Google Test or Robot Framework and their compatibility with eCAL.

(Optional) **How can integration tests be added to CI/CD workflows for automated validation?**
Focus on using CI tools like GitHub Actions to run tests and check results automatically.

Outline

1. **Introduction**
   - Role of IPC middleware in distributed systems
   - Motivation and research context
   - Objectives and scope of the thesis
   - Structure of the thesis
2. **Theoretical Background**
   - Fundamentals of distributed systems and IPC mechanisms
   - Middleware design and communication patterns (publish-subscribe, RPC)
   - Overview of eCAL and its core features
   - Comparison to other middleware frameworks (e.g., ROS, DDS)
   - Classification of testing strategies with focus on integration tests
3. **Requirements and Design of Integration Testing for IPC Middleware**
   - Analysis of middleware-specific test requirements
   - Communication scenarios: message loss, timing, multi-process flow
   - Functional and non-functional test objectives
   - Test strategy design: modularity, reusability, and coverage
4. **Tool Evaluation and Framework Selection**
   - Criteria for tool selection (language support, automation, IPC capabilities,)
   - Evaluation of candidates (e.g., Google Test, behave, Gauge, Robot Framework)
   - Feasibility study using eCAL
5. **Implementation of Integration Tests for eCAL**
   - Designing test architecture and environment
   - Creating test cases for representative IPC behaviors
   - Handling edge cases: timing jitter, unavailable publishers/subscribers
   - Automation setup for repeatable testing

6. **CI/CD Integration (Optional)**
   - Role of automated testing in software pipelines
   - Integrating the test framework into GitHub Actions
   - Test result visualization and reporting
7. **Evaluation and Discussion**
   - Coverage and effectiveness of the proposed test strategy
   - Observations from implementation and testing
   - Strengths, limitations, and lessons learned
8. **Conclusion and Outlook**
   - Summary of contributions
   - Applicability to other IPC frameworks
   - Suggestions for future extensions and research

Timeline and Approach

Timeframe	**Tasks**
March (Weeks 1-2)	Literature review and background research
March (Weeks 3-4)	Establishing the thesis structure, defining research scope
April (Weeks 1-2)	Familiarization with eCAL, analysis of existing tests, requirements gathering
April (Weeks 3-4)	Implementation of initial system tests, first validation
May (Weeks 1-2)	Testing in a distributed environment, evaluation of test effectiveness
May (Weeks 3-4)	CI/CD pipeline integration, refinement of testing strategies
June (Weeks 1-2)	Systematic evaluation of test results, further optimizations
June (Weeks 3-4)	Documentation of findings, finalizing thesis content
July (Weeks 1-4)	Final revisions, thesis presentation preparation

Bibliography

[1] Eclipse Foundation, eCAL - enhanced Communication Abstraction Layer, 2024. [Online]. Available: https://github.com/eclipse-ecal/ecal.

[2] A. Spillner and T. Linz, *Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester - Foundation Level - nach ISTQB -Standard*, 6th ed. Heidelberg: dpunkt.verlag, 2019.

[3] P. Liggesmeyer and S. Röttger, *Software-Qualität: Testen, Analysieren und Verifizieren von Software*, 2nd ed. Springer Vieweg, 2014.

[4] R. Bierig, S. Brown, E. Galván, and J. Timoney, "Introduction to Software Testing," in Essentials of Software Testing, Cambridge: Cambridge University Press, 2021.