Chapter 6

Real-Time Concurrency for Light-Weight Medical Robot Coordination in Shared Zones

Shu Kei Ling
The Hong Kong Polytechnic University, Hong Kong

Pak Sun Wong
The Hong Kong Polytechnic University, Hong Kong

Kwan Ho Yuen
http://orcid.org/0009-0007-8182-9222
The Hong Kong Polytechnic University, Hong Kong

Mohammad Al Khaldy
http://orcid.org/0009-0009-7502-4668
University of Petra, Jordan

ABSTRACT

This chapter is a Rust-based simulation system designed to coordinate multiple medical robots operating in shared hospital environments. The chapter focuses on concurrent systems, task scheduling, mutual exclusion for shared zones, and fault detection through heartbeat monitoring. A shared task queue assigns work to robot workers, while a zone access control mechanism ensures that only one robot operates in a critical area at a time. The system monitors heartbeat signals from each robot to detect failures or stalled execution. The modular architecture supports two execution models: a thread-based scheduler for lightweight concurrency and a process-based

DOI: 10.4018/979-8-2600-1101-0.ch006

scheduler for stronger isolation between workers. Controlled benchmarks and stress tests show correct enforcement of zone exclusivity, reliable detection of missed heartbeats, and stable performance under concurrent workloads. The results show predictable scalability for lightweight tasks and highlight contention effects when task duration and zone competition increase.

1. INTRODUCTION

1.1 Problem Statement and Motivation

Hospital operations involve constant movement of resources, personnel, and supplies under strict safety and timing constraints. Tasks such as delivering medication, transporting lab samples, moving equipment, and responding to emergencies often occur. When autonomous robots are introduced to assist with these operations, the coordination problem becomes more complex. The system must decide which robot performs which task, in what order, and under what safety guarantees.

Weak coordination can lead to serious issues. Multiple robots may attempt to enter the same corridor or treatment area at the same time, tasks may experience excessive delays and overall can lower the system efficiency even when sufficient robots are available. As a result. Robot coordination is not simply an execution problem, but a concurrency and scheduling problem.

A coordination system must handle or avoid dynamic task arrivals, shared resource conflicts, changes in robot availability, and occasional failures on design. Simple sequential schedulers are insufficient in this setting because they fail to protect shared state and are vulnerable to race conditions. At the same time, systems deployed in safety-critical environments must be observable and testable, allowing developers to validate behavior under both expected and extreme conditions.

The motivation for this chapter is to build a controlled simulation environment that models these challenges explicitly. Rather than treating concurrency as an implementation detail, Chapter Blaze places it at the center of system design. Task queues, schedulers, zone locks, and monitoring mechanisms (Mohammed & Zangana, 2026) are all modeled as first-class components. This approach allows the system to be evaluated not only on whether tasks complete, but on how safely and efficiently they complete under varying levels of load. In addition, the chapter serves an educational objective: applying systems programming concepts to a realistic domain where synchronization, isolation, and fault awareness are essential rather than optional.

Designing Mobile Learning Smart Education System Architecture for Big Data Management Using Fog Computing Technology

Muhammad Adnan Kaim Khani, Abdullah Ayub Khan, Allah Bachayo Brohiand Zaffar Ahmed Shaikh (2022). *The International Journal of Imaging and Sensing Technologies and Applications (pp. 1-23).*

www.irma-international.org/article/designing-mobile-learning-smart-education-system-architecture-for-big-data-management-using-fog-computing-technology/306653