# Summary


This Ph.D. thesis investigates what is required to build a coupled, perpetual, and dynamic maintenance scheduling system for offshore oil and gas operations. The work draws on literature from three domains: maintenance scheduling, software engineering, and mathematical optimization.

The research followed the design science method, iterating through problem definition, requirements gathering, design and development, demonstration, evaluation, and communication—aligning the mathematical optimization approaches with the requirements of decision-making stakeholders.

The scheduling system needed three properties to function in practice:

**Coupled.** The maintenance scheduling domain consists of four separate but interdependent schedules: project, weekly, daily, and operational. Each schedule type can be instantiated multiple times and must remain consistent with the others.

**Perpetual.** Schedules must be continuously available to decision makers. A perpetual system ensures that coupled schedules remain mutually consistent and eliminates coordination overhead regarding when to run the scheduling program.

**Dynamic.** Decision makers must be able to modify the system without restarting, incorporate tacit knowledge into their schedules, and integrate with live data sources such as ERP systems.

These requirements led to three engineering contributions, each disseminated as a journal paper:

1. **Actor-based Large Neighborhood Search for Weekly Maintenance Scheduling.** Demonstrates that perpetual and dynamic metaheuristics can extend proven optimization approaches. Key result: enables dynamic modification of resources, work order scheduling and unscheduling, and priority changes for a quadratic multi-compartment multi-knapsack problem during optimization.

2. **Read-Copy-Update Pattern for Parallel Metaheuristics.** Contributes a paradigm for designing parallel optimization systems that scale in practice. Key result: Parallelize 4 types of metaheuristics in up to 33 coupled instances.

3. **Hypergraph as Domain Model for Coupled Scheduling Systems.** Addresses domain modeling challenges in both mathematical optimization and software engineering, where conventional practices struggle with complex scheduling problems. Key result: demonstrates that a hypergraph can model complex intra- and inter-schedule constraints for up to 1,000 work orders, 100 technicians, over 730 days.

Several observations emerged during the research: (i) low code sharing in the literature impedes verification and collaboration, particularly for practical decision systems that are built for specific enterprises; (ii) heuristics can optimize decision problems richer in detail than current modeling notations support; (iii) systems-level programming (C, C++, Rust) is fundamental for effective parallel optimization, as memory layout and pointer semantics become critical when transitioning from sequential to parallel environments.

Further research is needed to verify the effectiveness of the read-copy-update pattern at scale and to determine whether hypergraphs are a hard requirement for coupled and dynamic scheduling systems or merely a parsimonious conceptual model.