SOFTENG 752
Formal Specification and Design
Assignment #2
A Summary Discussion of the B, OCL and CSP Modelling

Name: Enyang Zhang
UPI: ezha812
ID: 721492553

Assignment partner: Jennifer Xie
UPI: yxie614
ID: 556477746

1. Overview

This project extends the formal specification from A1 by adding two abstraction layers—UML + OCL and CSP—to capture both structural integrity and dynamic behaviour of the Library Management System. The B model provided a rigorous mathematical basis through state variables, invariants, and operations. The OCL model refined these concepts into an object-oriented structure with constraints validated in USE, while CSP translated verified state changes into event-driven processes. Together, the three models form a continuous refinement chain from logical specification to structural and behavioural verification.

2. Connections in Design

The three models were developed sequentially after analysing the functional requirements of a library domain, each enhancing the prior iteration while maintaining semantics. In B, invariants established logical relations such as "a reserved book cannot be borrowed" and "waiting members are distinct." These requirements were subsequently reflected in OCL, where we modeled the BookCopy, Member, and Loan classes and validated preconditions and postconditions such as borrowBook and returnBook in USE. The validated OCL operations are then transformed into CSP events—borrow → getBook, returnBook → leaveBook—to represent concurrent interactions between users and admins. This approach guaranteed that any behavioral trace in CSP could be linked to a properly proven state rule in B.

3. Comparative Analysis: Pros and Cons

Each formalism provided unique advantages to the comprehensive specification. The B technique provided mathematical accuracy and facilitated early error identification, although necessitated extensive notation and was less intuitive for structure reasoning. OCL enhanced readability by object-oriented abstraction and visual validation in USE, while it did not provide explicit support for concurrency or temporal behavior. CSP bridged that gap by modeling communication and synchronization, allowing ProB to test deadlock freedom and interaction accuracy while abstracting low-level data semantics. The three models collectively ensured rigor, clarity, and executability: B provided logical soundness, OCL upheld structural consistency, and CSP guaranteed behavioral reliability, creating a complementary toolbox for producing a verifiable and intelligible system design.

4. Lessons Learned

During the development phase, each formalism revealed distinct practical challenges. In B, establishing the invariant for the waiting queue—ensuring that no member appeared twice and that reservations consistently aligned with availability—was subject to logical errors that resulted in ProB violations. Translating to OCL, the representation of the same rule via Sequence(Member) introduced uncertainty in multiplicities and navigation from BookTitle to BookCopy. In CSP, the synchronisation of borrow and returnBook events between the member and admin processes initially resulted in concealed deadlocks. Addressing these challenges demonstrated that cross-model consistency requires intentional refinement and ongoing semantic verification, rather than simple translation.

5. Conclusion

The integration of B, OCL, and CSP established a consistent pathway from abstract reasoning to executable behavior. Through the iterative refinement of a singular design across many formalisms, we achieved enhanced assurance of both correctness and consistency. The method illustrated that formal specification is not a single task but a progressive framework for creating reliable systems with clear logic and verifiable interactions.