IMPLEMENTING AND VERIFYING THE SAFETY OF THE TRANSACTOR MODEL

By

Brian Boodman

An Abstract of a Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Major Subject: COMPUTER SCIENCE

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Approved:

Carlos Varela, Thesis Adviser

Rensselaer Polytechnic Institute
Troy, New York

April 2008
(For Graduation May 2008)

ABSTRACT

The transactor model is an extension of the actor model designed to tolerate failures in distributed systems. Transactors can provide guarantees about consistency of a distributed system's state in the face of message loss and temporary failures of computing nodes. The model introduces dependency information and a two-phase checkpointing protocol. The added dependency information enables transactors to track the interdependencies caused by communications between actors, making it possible to ensure that the state of the distributed program as a whole remains globally consistent. This thesis discusses the use of three tools in order to test and prove the safety of the *transactor* model. We used Maude rewrite systems as a tool to test the model behavior and to discover problems with the model. During this stage, we discovered a safety bug and proposed changes to fix it. We then used the Athena proof verification system to show that the updated model is safe. Finally, we used the Salsa actor programming language as a basis for a higher-level transactor-based prototype programming language.

First, we developed a prototype implementation of the transactor model using Maude. Maude's underlying rewriting rules system is well-suited towards developing an executable operational semantics for concurrent programming models. The implementation was used to test example programs and check the transactor model's safety. This prototype was in fact used to discover a safety error.

Subsequently, we wrote a formal proof in the Athena language. As a multi-sorted first order logic system, Athena provides an effective means of representing the transactor model's correctness properties. Because Athena proofs are computer-checkable, they are more reliable than traditional proofs. Further, Athena permits the use of automated theorem proving, allowing us to skip tedious steps which would otherwise unnecessarily complicate the proof's readability.

Finally, we developed a coordination language using Salsa and Java. The language provides a practical demonstration of the use of the transactor model and shows some of the potential issues in creating an effective implementation of the

model.