

# Mission Critical Systems: Project Proposal

**Due:** March 1, 2016

**Name:** Skyler Layne

---

| Revision | Date         |
|----------|--------------|
| 1.0      | 4 March 2016 |

---

## Overview

I would like to extend the JPF model checker to add functionality for database accesses. While Java has the functionality to interact with a database, some of these methods are native methods. This makes it difficult to model check with JPF as, in general, JPF does not play well with native methods. I.e. additional functionality is needed to allow JPF to connect to the database, and continue model checking.

## Objective

This project proposal includes an extension of the JPF model checker to add functionality to account for the native methods used by Java in accessing databases. In this extension I would like to focus on the MySQL database as this is where I have the most experience, but I am open to exploring a solution with an In-memory database such as H2. Java has functionality to interact with a MySQL database through native methods which are implemented in the JDBC (Java Database Connectivity) package. In general, JPF does not play well with these or any other native methods.

I propose the implementation of a native peer to communicate with the database through a model class which holds the database connection. Given my understanding of JPF in regards to native peers and model classes, this will allow JPF to reach into the JVM and execute the native methods without dropping or losing connection to the database as desired.

This is an interesting topic to me because many applications rely on interacting with a database; this can and will fail. I believe, we can use model checking techniques to catch these potential errors before they surface.

Similar work has been done in this field, specifically in the paper *Model Checking Database Applications* [1]. The solution Gligoric and Majumdar, come up with has a strong focus on modeling the data reliability between a Java application and an in-memory database (H2). They examine multithreaded web applications to ensure that the data is preserved. Similar to this project I would like to allow JPF to look into a database, however, I would not like to concern myself as thoroughly with data reliability as they have. I would like to focus purely on allowing JPF to execute instructions on a database, i.e. carryout the native methods which are defined in the JDBC package.

## Goals

The target goals that I would like to reach include, but are not limited to:

- Implement a model class to hold the MySQL connection.
- Implement native peer for communication with the database.

## References

- [1] *Model Checking Database Applications* by Milos Gligoric and Rupak Majumdar.