At MongoDB, we use TLA+ to specify multiple aspects of several systems. We develop specifications and implementations concurrently, and it would be valuable to us if we could continuously test that our specifications and implementations conform. The "eXtreme Modelling" methodology[1] proposes techniques for testing conformance; we performed two case studies to experiment with two of them.

In the first case study, we attempted to test the conformance of the MongoDB Server to its TLA+ specifications using *model-based trace-checking*. We captured execution traces from the server as we fuzz-tested it, and checked that these traces were permitted by the spec. This case study ended in failure: the project grew to cost more effort than it was worth, and we cancelled it. I will describe three factors that led to our failure and what we would do differently if we tried again.

In the second case study, we tested the conformance of MongoDB Realm Sync to its spec using *model-based test-case generation*. We enumerated all behaviors of the spec, and generated C++ unit tests to check that the implementation conforms to each. This project was successful and achieved 100% branch coverage of the specified algorithm. I will compare this case study to the model-based trace-checking case study, and explain why this project was successful.

This presentation will be useful to engineers who specify industrial software systems. They will learn about two useful testing techniques, and our story may help them repeat our success and avoid repeating our mistakes.

Thank you for considering,
Jesse

[1] A. M. Gravell, Y. Howard, J. C. Augusto, C. Ferreira, and S. Gruner. Concurrent Development of Model and Implementation. CoRR, abs/1111.2826, 2011.