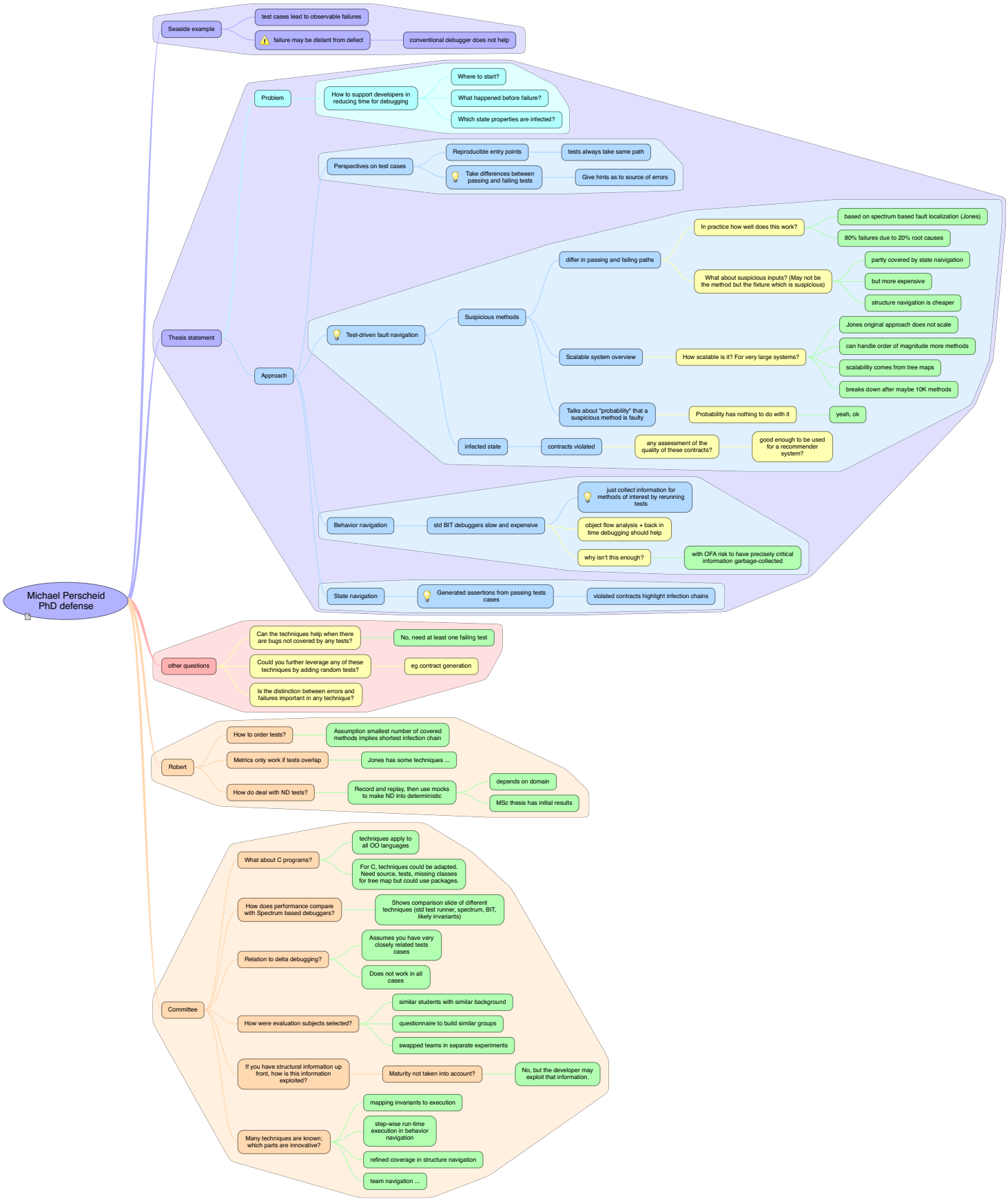
Michael Perscheid
PhD defense
Seaside example
test cases lead to observable failures
failure may be distant from defect
conventional debugger does not help
Thesis statement
Problem
How to support developers in reducing time for debugging
Where to start?
What happened before failure?
Which state properties are infected?
Perspectives on test cases
Reproducible entry points
tests always take same path
Take differences between passing and failing tests
Give hints as to source of errors
Approach
Test-driven fault navigation
Suspicious methods
differ in passing and failing paths
In practice how well does this work?
based on spectrum based fault localization (Jones)
80% failures due to 20% root causes
What about suspicious inputs? (May not be the method but the fixture which is suspicious)
partly covered by state naivigation
but more expensive
structure navigation is cheaper
Scalable system overview
How scalable is it? For very large systems?
Jones original approach does not scale
can handle order of magnitude more methods
scalability comes from tree maps
breaks down after maybe 10K methods
Talks about "probability" that a suspicious method is faulty
Probability has nothing to do with it
yeah, ok
infected state
contracts violated
any assessment of the quality of these contracts?
good enough to be used for a recommender system?
Behavior navigation
std BIT debuggers slow and expensive
just collect information for methods of interest by rerunning tests
object flow analysis + back in time debugging should help
why isn't this enough?
with OFA risk to have precisely critical information garbage-collected
State navigation
Generated assertions from passing tests cases
violated contracts highlight infection chains
other questions
Can the techniques help when there are bugs not covered by any tests?
No, need at least one failing test
Could you further leverage any of these techniques by adding random tests?
eg contract generation
Is the distinction between errors and failures important in any technique?
Robert
How to order tests?
Assumption smallest number of covered methods implies shortest infection chain
Metrics only work if tests overlap
Jones has some techniques ...
How do deal with ND tests?
Record and replay, then use mocks to make ND into deterministic
depends on domain
MSc thesis has initial results
Committee
What about C programs?
techniques apply to all OO languages
For C, techniques could be adapted. Need source, tests, missing classes for tree map but could use packages.
How does performance compare with Spectrum based debuggers?
Shows comparison slide of different techniques (std test runner, spectrum, BIT, likely invariants)
Relation to delta debugging?
Assumes you have very closely related tests cases
Does not work in all cases
How were evaluation subjects selected?
similar students with similar background
questionnaire to build similar groups
swapped teams in separate experiments
If you have structural information up front, how is this information exploited?
Maturity not taken into account?
No, but the developer may exploit that information.
Many techniques are known; which parts are innovative?
mapping invariants to execution
step-wise run-time execution in behavior navigation
refined coverage in structure navigation
team navigation ...

Michael Perscheid PhD defense. HPI, Potsdam, 2013-09-20
Test-driven Fault Navigation for Debugging Reproducible Failures