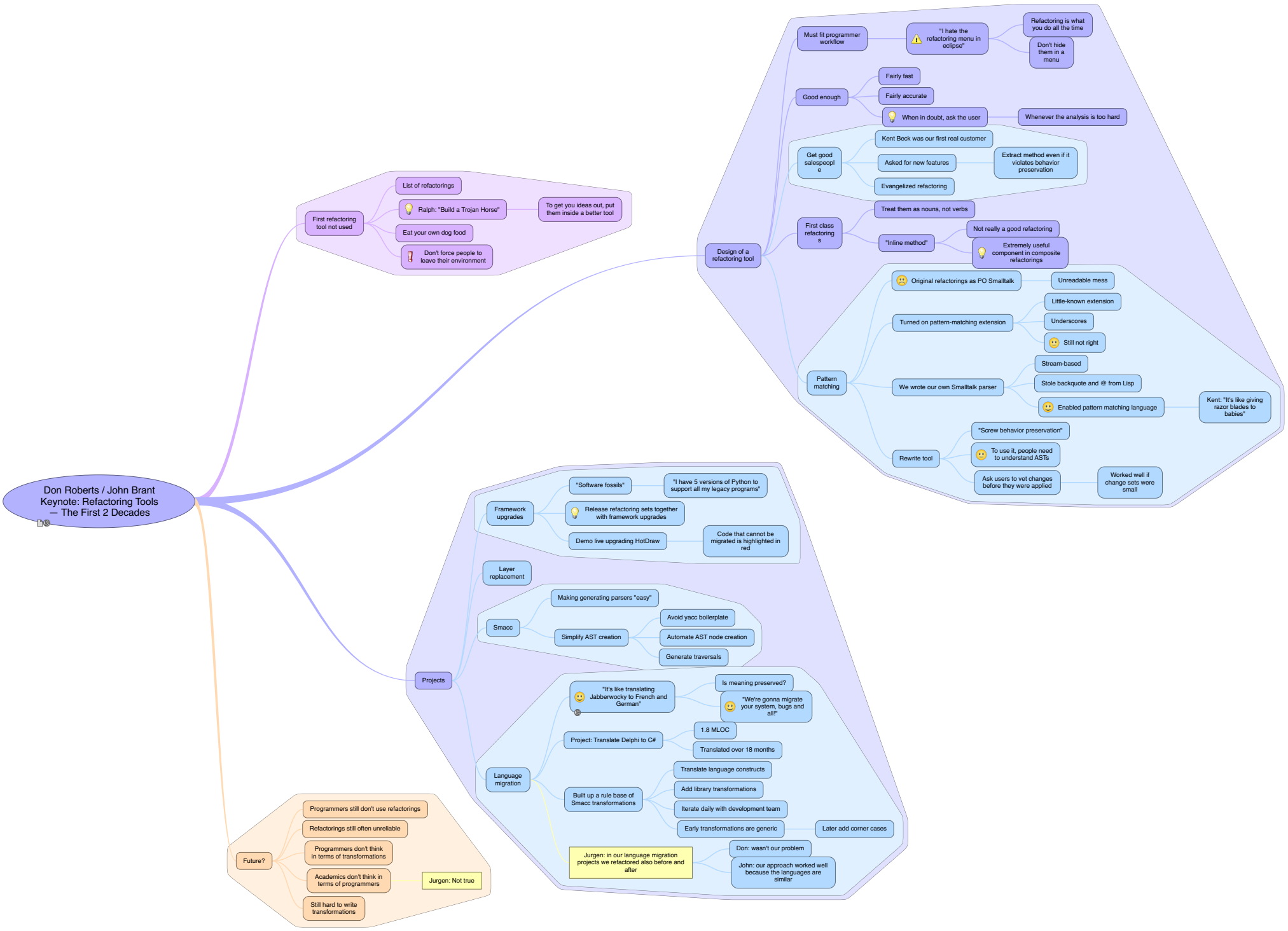
Don Roberts / John Brant
Keynote: Refactoring Tools
— The First 2 Decades
First refactoring tool not used
List of refactorings
Ralph: "Build a Trojan Horse"
To get you ideas out, put them inside a better tool
Eat your own dog food
Don't force people to leave their environment
Design of a refactoring tool
Must fit programmer workflow
"I hate the refactoring menu in eclipse"
Refactoring is what you do all the time
Don't hide them in a menu
Good enough
Fairly fast
Fairly accurate
When in doubt, ask the user
Whenever the analysis is too hard
Get good salespeople
Kent Beck was our first real customer
Asked for new features
Extract method even if it violates behavior preservation
Evangelized refactoring
First class refactorings
Treat them as nouns, not verbs
"Inline method"
Not really a good refactoring
Extremely useful component in composite refactorings
Pattern matching
Original refactorings as PO Smalltalk
Unreadable mess
Turned on pattern-matching extension
Little-known extension
Underscores
Still not right
We wrote our own Smalltalk parser
Stream-based
Stole backquote and @ from Lisp
Enabled pattern matching language
Kent: "It's like giving razor blades to babies"
Rewrite tool
"Screw behavior preservation"
To use it, people need to understand ASTs
Ask users to vet changes before they were applied
Worked well if change sets were small
Projects
Framework upgrades
"Software fossils"
"I have 5 versions of Python to support all my legacy programs"
Release refactoring sets together with framework upgrades
Demo live upgrading HotDraw
Code that cannot be migrated is highlighted in red
Layer replacement
Smacc
Making generating parsers "easy"
Simplify AST creation
Avoid yacc boilerplate
Automate AST node creation
Generate traversals
Language migration
"It's like translating Jabberwocky to French and German"
Is meaning preserved?
"We're gonna migrate your system, bugs and all!"
Project: Translate Delphi to C#
1.8 MLOC
Translated over 18 months
Built up a rule base of Smacc transformations
Translate language constructs
Add library transformations
Iterate daily with development team
Early transformations are generic
Later add corner cases
Jurgen: in our language migration projects we refactored also before and after
Don: wasn't our problem
John: our approach worked well because the languages are similar
Future?
Programmers still don't use refactorings
Refactorings still often unreliable
Programmers don't think in terms of transformations
Academics don't think in terms of programmers
Jurgen: Not true
Still hard to write transformations

/ Don Roberts / John Brant Keynote: Refactoring Tools — The First 2 Decades

Dagstuhl 14211 - May 22, 2014
http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=14211

/ Don Roberts / John Brant Keyno… / Projects / Language migration / "It's like translating Jabberw…

http://en.wikipedia.org/wiki/Jabberwocky#Translations