Reengineering with Reflexion Models: A Case Study

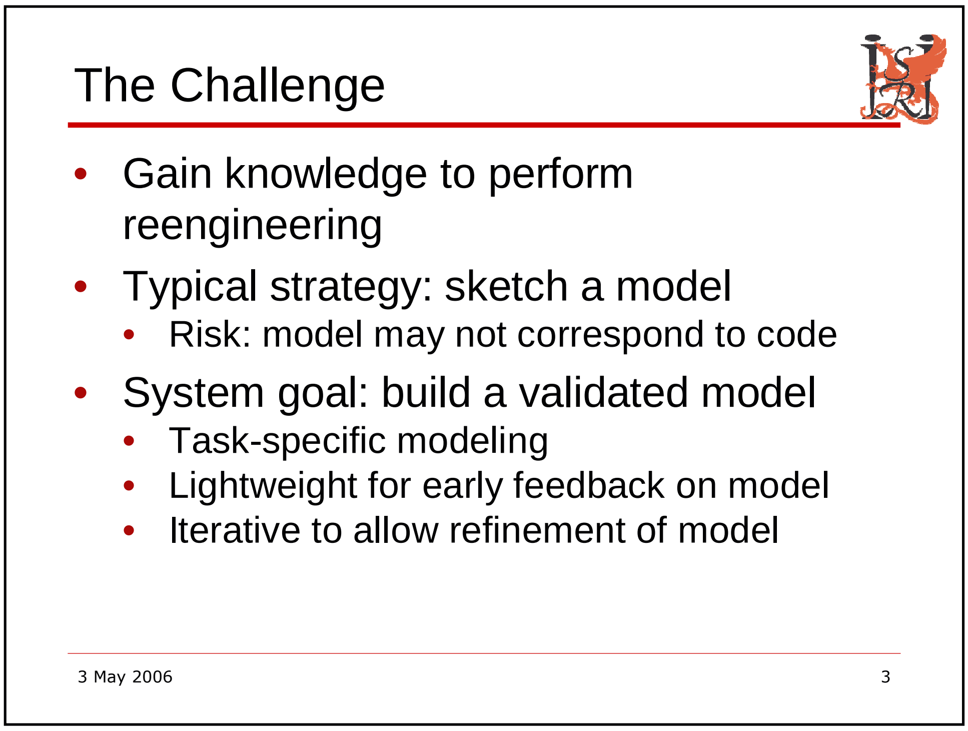
Based on the IEEE Computer article by Gail Murphy and David Notkin

17-654/17-754: Analysis of Software Artifacts

Jonathan Aldrich

Task

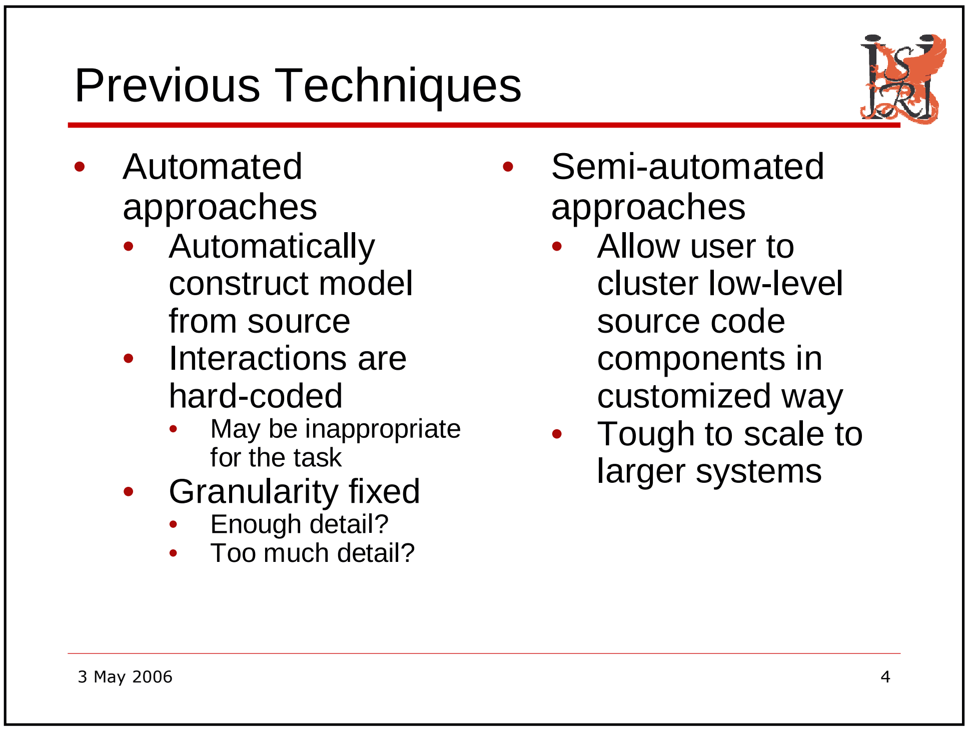
- Reengineer Excel code
  - 1.2 million LOC
  - Extract components



The Challenge

- Gain knowledge to perform reengineering
- Typical strategy: sketch a model
  - Risk: model may not correspond to code
- System goal: build a validated model
  - Task-specific modeling
  - Lightweight for early feedback on model
  - Iterative to allow refinement of model

3 May 2006 3



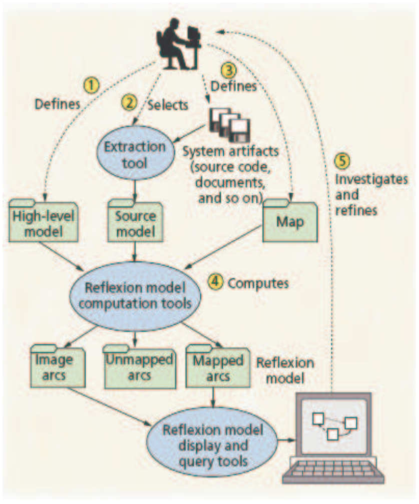
Previous Techniques

- Automated approaches
  - Automatically construct model from source
  - Interactions are hard-coded
    - May be inappropriate for the task
  - Granularity fixed
    - Enough detail?
    - Too much detail?
- Semi-automated approaches
  - Allow user to cluster low-level source code components in customized way
  - Tough to scale to larger systems

3 May 2006 4

Basic Approach

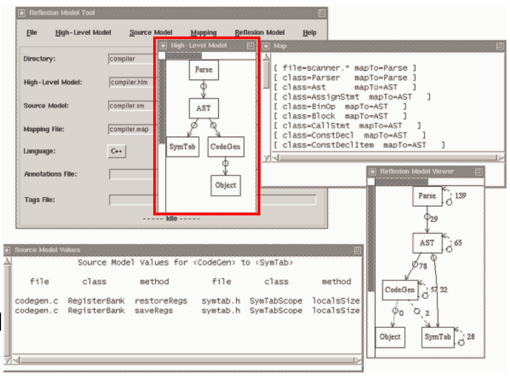
- Hypothesize a Model
- Describe mapping to code
  - Can use tools customized to task
- Validate model vs. code
  - Tool shows differences
- Refine model and/or mapping and iterate



① Defines
② Selects
③ Defines
Extraction tool
System artifacts (source code, documents, and so on)
⑤ Investigates and refines
High-level model
Source model
Map
Reflexion model computation tools
④ Computes
Image arcs
Unmapped arcs
Mapped arcs
Reflexion model
Reflexion model display and query tools

Defining a Model

- Graph of nodes and arcs
- Based on knowledge, documentation, or browsing code
- Can be arbitrary
- Estimate: 15-60 min



```
[ file=scanner.* mapTo=Parse ]
[ class=Parser   mapTo=Parse ]
[ class=Ast      mapTo=AST   ]
[ class=AssignStmt  mapTo=AST   ]
[ class=BinOp  mapTo=AST   ]
[ class=Block  mapTo=AST   ]
[ class=CallStmt  mapTo=AST   ]
[ class=ConstDecl  mapTo=AST   ]
[ class=ConstDeclItem  mapTo=AST   ]
```

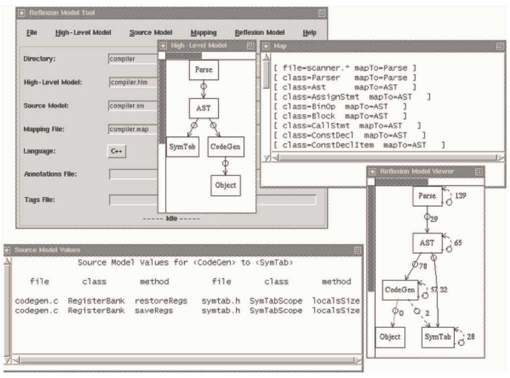
```
              Source Model Values for <CodeGen> to <SymTab>

  file        class          method        file       class          method

codegen.c   RegisterBank   restoreRegs   symtab.h   SymTabScope   localsSize
codegen.c   RegisterBank   saveRegs      symtab.h   SymTabScope   localsSize
```

Extract Source Model

- Use tool
- Excel example
  - Call-graph constructor
  - Approximates desired dataflow information
  - Could be different in other applications
- Shows dependences between source elements



Directory: compiler
High-Level Model: compiler.hlm
Source Model: compiler.sm
Mapping File: compiler.map
Language: C++
Annotations File:
Tags File:
----- Idle -----

```
[ file=scanner.* mapTo=Parse ]
[ class=Parser   mapTo=Parse ]
[ class=Ast      mapTo=AST   ]
[ class=AssignStmt  mapTo=AST   ]
[ class=BinOp  mapTo=AST   ]
[ class=Block  mapTo=AST   ]
[ class=CallStmt  mapTo=AST   ]
[ class=ConstDecl  mapTo=AST   ]
[ class=ConstDeclItem  mapTo=AST   ]
```

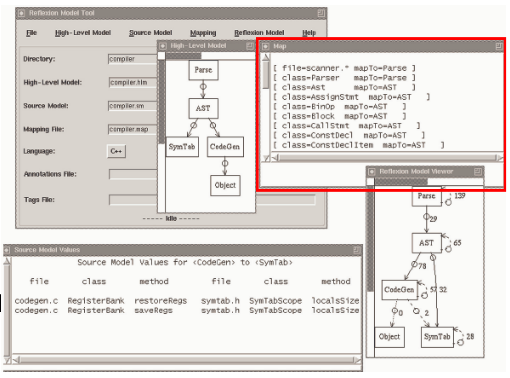
```
            Source Model Values for <CodeGen> to <SymTab>

  file        class         method          file       class         method

codegen.c   RegisterBank   restoreRegs    symtab.h   SymTabScope   localsSize
codegen.c   RegisterBank   saveRegs       symtab.h   SymTabScope   localsSize
```

Defining a Mapping

- Source elements to nodes
  - Files
  - Classes
  - Functions
- First cut may be approximate
- Estimate: 10-30 min


Directory: compiler
High-Level Model: compiler.hlm
Source Model: compiler.sm
Mapping File: compiler.map
Language: C++
Annotations File:
Tags File:
----- Idle -----

```
[ file=scanner.* mapTo=Parse ]
[ class=Parser   mapTo=Parse ]
[ class=Ast      mapTo=AST   ]
[ class=AssignStmt  mapTo=AST   ]
[ class=BinOp  mapTo=AST   ]
[ class=Block  mapTo=AST   ]
[ class=CallStmt  mapTo=AST   ]
[ class=ConstDecl  mapTo=AST   ]
[ class=ConstDeclItem  mapTo=AST   ]
```

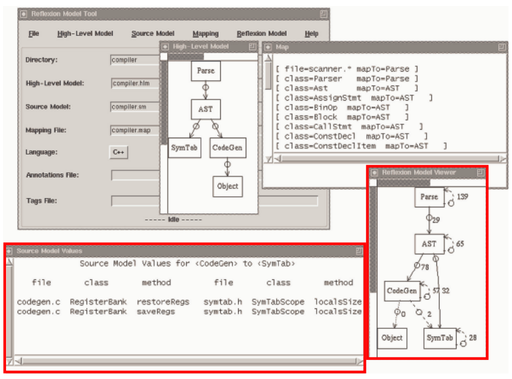
```
            Source Model Values for <CodeGen> to <SymTab>

  file        class         method          file       class         method

codegen.c   RegisterBank   restoreRegs    symtab.h   SymTabScope   localsSize
codegen.c   RegisterBank   saveRegs       symtab.h   SymTabScope   localsSize
```

Reflexion Model

- Shows high-level model
  - Convergences
  - Divergences
  - Absences
- Can investigate arcs
  - Provides valuable information for refining model



```
[ file=scanner.* mapTo=Parse ]
[ class=Parser   mapTo=Parse ]
[ class=Ast      mapTo=AST   ]
[ class=AssignStmt  mapTo=AST   ]
[ class=BinOp  mapTo=AST   ]
[ class=Block  mapTo=AST   ]
[ class=CallStmt  mapTo=AST   ]
[ class=ConstDecl  mapTo=AST   ]
[ class=ConstDeclItem  mapTo=AST   ]
```

```
              Source Model Values for <CodeGen> to <SymTab>

   file       class        method        file       class        method

codegen.c  RegisterBank  restoreRegs   symtab.h  SymTabScope  localsSize
codegen.c  RegisterBank  saveRegs      symtab.h  SymTabScope  localsSize
```

Refinement Process

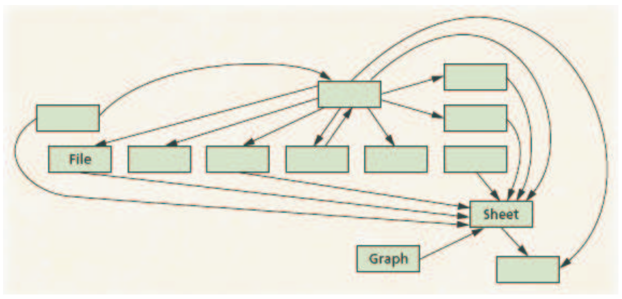
- Address divergences/absences
  - Modify model
  - Modify mapping
    - e.g., function g belongs in file f, but was in file p instead
    - Tendency to add functions where the cursor is!
- Refine model
  - Split a node into parts, specify substructure

State of Excel Documentation

Excel Internals . . . explains the philosophy of a few of the basic things in Excel, like the cell table formulas, memory allocation, a little bit about the layer [a special interface with the operating system that allows Microsoft to use the same Excel core on both Windows and Macintosh platforms]. . . . It's very sparse. We don't necessarily rely on that for people to learn things. I'd say we have a strong oral tradition, and the idea is that the mentor teaches people or people learn it themselves by reading code. . . . Over the course of a project, it goes from mostly truthful to less truthful, and then we have to fix it up. We don't fix it up as we go along on a project. We will give it some attention between projects.

3 May 2006 11

Initial Modeling Process



- Reading Excel Internals
- Brief discussion with team members
- Drew "natural" model

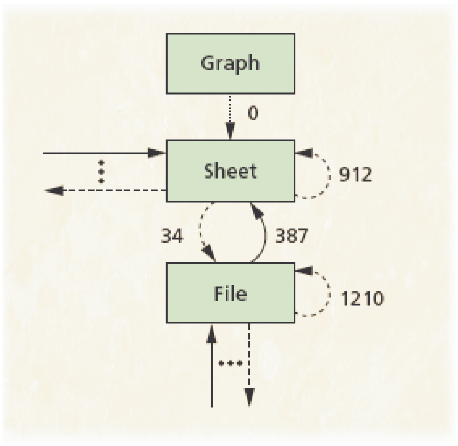
3 May 2006 12

Source Model and Mapping

- Source Model constructed by internal Microsoft call-graph building tool
  - 77,746 calls!
- Mapping
  - 170 lines long
  - Describes 400 files
  - Took a few hours

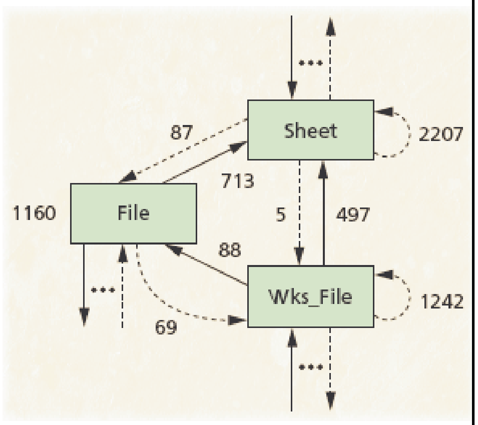
Initial Reflexion Model

- Tasks
  - Update model
    - if reasonable interactions missing
  - Investigate edges
    - to learn about source
  - Update map
    - exceptions for functions logically in another module
    - ultimately 1000 lines long
  - Extended source model
    - global variables
- Detailed focus on relevant parts of system
- Work done with scripts



Resulting Model

- Benefits
  - Understanding of system
    - Unexpected dependences
  - Feasibility of reengineering
    - How many arcs would be cut?
  - Aid in component isolation
    - Insert conditional compilation based on map



Reengineering Tool: Lessons Learned

- Task-specific views are important
  - Developer didn’t want to waste time on irrelevant parts of the system
- Connection to code important
  - Both for understanding and for reengineering task itself
- Both text and GUI interfaces needed
  - Most real work done with text!
- Adaptable tools needed
  - Engineer wrote scripts to process input/output files