Analysis of Software Artifacts

Reverse Engineering Case Study (Lecture 17)
Reverse Engineering

- Local optimizations to code
- Incomplete understanding about the program
- Various coding styles
- Different programmers
  - "worked on" by a lot of programmers
  - Frequently code over a long period of time
code decay disease

- compiler optimizers notoriously suffer from the
  usually happens to very intricate pieces of code

philosophy

- usually because there isn't a consistent design

complex

- the structure of the program becomes very

leads to code decay

Reverse Engineering
The Reverse Engineering Process

- The model stabilizes
- Iterate between the two steps until
- Use program understanding
- Build a model of what the system does

Model building

- Use program analysis tools
- Talk to the software architects
- Use existing documentation (if you are lucky!)

Program understanding

________________________________________

The Reverse Engineering Process
The Forward Engineering Process

- Frequently have to refine the model
- Test the code and iterate the process
- Engineer legacy COBOL code
- Large effort going on in the industry to reverse (C, C++, Java)
- Use a modern programming language
- Use the model to construct the code
- Assume that the model has been constructed
communication manager
re-engineered a piece of the CTAS system called the

twelve students and three faculty

Graduate seminar at MIT
case study was a one-semester

The CTAS Case Study
What is CTAS?

percent

(DFW) Terminal improved landing rate by 10%

• a prototype deployed at the Dallas/Fort Worth

• automated planning

• purpose is to increase landing rate through

• flow

suite of tools to help controllers manage air traffic

__________________________

What is CTAS?
What is CTAS?

- Predicts aircraft trajectories as much as forty minutes in advance
- Minimizes unused landing slots
- Uses this data to suggest a landing sequence to
- Commands standard landing patterns and controller
- Weather data and available runways
- Location, velocity, and flight plans
- Receives input about
What is CM?

- maintains the database of aircraft information
- components
- acts as a message switch moving data among center
- The Communications Manager (CM) sits at the center
What is CM?

- 80,000 lines of C and C++ code
- Single point of failure in the entire system
- Repository for several unrelated features
- CM has become very complex
- CTAS system
- Basically is the communication hub for the entire
What was done?

• Just the core functionality of CM
• Did not implement the entire CM
• Re-implemented CM in JAVA
• Built a model of the CM
  used program understanding to
node represents a function

Imagine a graph where each

particular function

what are the functions that call or are called by a

syntactic information about programs

these tools are used to find

____________________________________

Syntactic Tools
used calls between files very effectively

- MIT students used a tool called ImagX
  - the graph is called a call graph
  - there is an edge from f to g if f calls g

---

Syntactic Tools
directly or indirectly?

What are the statements that effect a certain record?

- Data-flow analysis

- For example, syntactic tools do not infer any semantic information

- Syntactic tools do not infer any semantic information

Semantic Tools
Program slicing tool

- http://www.grammatech.com

- check out the CodeSurfer homepage at

- will talk about both the tools in the next lecture

- in the MIT project they used LackoFit and CodeSurfer

Semantic Tools
Other Tools Developed

- Optionally a two line specifications
- List of calling and called functions
- Arguments and results
  - for each function stored
- Program understanding
- Web-interface to information gathered by
  Concordance Generator
a specialized script for handling messages •

CM

visualize the flow of messages that happen in the
message sequence charts

a post-processor to convert message traces into
message sequence chart generator •

Other Tools Developed
Problems Found

- Monitoring
  - FAA would like to monitor the workings of CTAS
  - adding this to existing CM was difficult

- CM goes down the entire CTAS crashes

- System was not fault-tolerant

- Could cause CM to deadlock

- Blocking Sends
  - Sending of messages uses blocking primitives
Lessons Learned

- Where should education cycles be spent?
- Education go debate about where software engineering
- Data abstraction – standard software engineering techniques work
- Simple designs are possible

---
Lessons Learnt

- Constructing object models
- High level models are vital
- Reverse engineering tools work
- Consistent commenting criteria
- Tools
- Good coding style helps the program analysis
- Rigorous coding standards
- Reverse engineering efforts helped by NASA's coding standards are vital

Lessons Learnt