

### Capturing a Holistic View from a Legacy Code Base

#### Table of Contents



2	Architecture Overview	



4 Open Problems
-----------------

#### Background

- Software out sourcing business
- Legacy system maintenance
- System reengineering
- Wanted improved productivity

#### Vision

- BlueMorpho Project
- Multi-language, multi-platform
- Code Analysis
- Business Rule Extraction
- Language Translation
- Architecture Transformation

## Reengineering is about bridging 2 pyramids with a holistic view



#### Architecture Overview





#### • A 28K-line COBOL program

- Mapping hundreds of fields between 2 data structures
- Hard to maintain

#### o Analysis Results

- Complex call graph
- Many duplicate code
- Uninitialized fields

Dup Blocks	4	195	2	9	5	3	2	9	2	7	3	3	184	10	0
Dup Lines	4	913	л	15	31	з	2	61	53	7	45	45	827	27	0
Dup Lines%	E 200/	49.40%	10.42%	41.67%	46.97%	3.19%	5.71%	98.39%	100.00%	36.84%	100.00%	100.00%	48.03%	100.00%	b/T 0'C 7
	-														

#### • Rewrite guide lines

- Field by field mapping, instead of conditional branches
- Map each field once, no overrides

#### Code Analysis - Sample views



- Slicing code
  - Pattern Matching
- B.R. description
  - Recommendation



#### BRE – Case Study

- COBOL programs with many "edits"
- Different combination of "edits" for each member
- Our results
  - Edits distribution / combination matrix
  - Recommending description based on code similarity

Para	Code	Edit Description	00000000	Valation	Vercento
1100	IF WE UNDERSON THE OR WE CONTENT THE	if drug on UN250000 or CC250007 set			
	MOVE 'L' TO DEDUCT-OPT	deduct option to L.			
	END-IF		*		*
•••	•••	•••			



- 100+ COBOL "View Calls"
  - Data query programs
  - EXEC SQL statements

#### o Approach

 Rule development driven by language feature and errors

COBOL	LOC of COBOL	СОРҮВО ОК	LOC of copybook	Generated Java files	LOC of Java	Compile errors	E) ra	xpansion Ite of files	Expansion rate of LOC
111	206810	560	28674	2158	576763		0	321.61%	244.93%

#### o Results

- 3 months rule correction / customization
- 100% translated, zero compile error
- Selected 5 for validation
  - Side by side output comparison with COBOL pass
  - Manual SIT test with web front end pass



- A sample Java EE application
- Apply several transformations to utilize "cloud services" on AWS

3

Replacing local databas

```
4
                                              5
                                                    $newFile*
 1 import "avoidUsingJavaFileClass.pat"
                                              6
 2
                                              7
   match "#avoidUsingJavaFileClass"
 3
                                              8
                                                    $*
 4
                                              9
 5 delete "begin"
                                             10
 6
                                             11
                                                    $*
   replace "deleteExists" do
 7
                                             12
     use_template "deleteS30bject.ftl"
 8
                                             13
 9
   end
                                             14
                                                    $*
   replace "writeFile" do
10
                                             15
     use_template "putObject.ftl"
11
                                             16 #);
12 end
13 replace "params" do
     use_template "\"posterKey\", savePath+name+extName"
14
15 end
16
  replace "path" do
17
     use_template "newS3Client.ftl"
18
19 end
20
   delete "newFile"
21
22
23
```

```
1 avoidUsingJavaFileClass = (#
2 $returnType $methodName( $* ) $* {
     $begin*
     $path(#"/$*"#)
   $deleteExists(# if(request $*){ $* }#)
     $writeFile(# do{ $* }while($*) $* try{ $* } catch($*){ $*}#)
     (# setAttribute( $params* ) #)
```

ification on multiple

#### o A holistic view on legacy code base is critical in

- Maintenance
  - What do these programs do, and how
  - Where are the major problems
- Reengineering
  - What logics to preserve and what not
  - Is there a repeated pattern
- A holistic view is consist of
  - A system model of code artifacts and interconnections
  - A business model of concepts and workflow

#### **Open Problems**

#### o Efficient Semantic Model Storage

- General enough (any objects + any relations; RDF preferred)
- Fast Query (Graph databases?)
- Incremental update on code

#### o Effective Machine Learning / NLP

- Business model from documents
- Annotation recommendation
- Natural language description (SMT?)

#### **o** Visualization

- Large model diagrams
- Interactive model navigation

#### • Dynamic Analysis

• Unify with static model

# THANK YOU