ISP RAS
Institute for System Programming
of the Russian Academy of Sciences

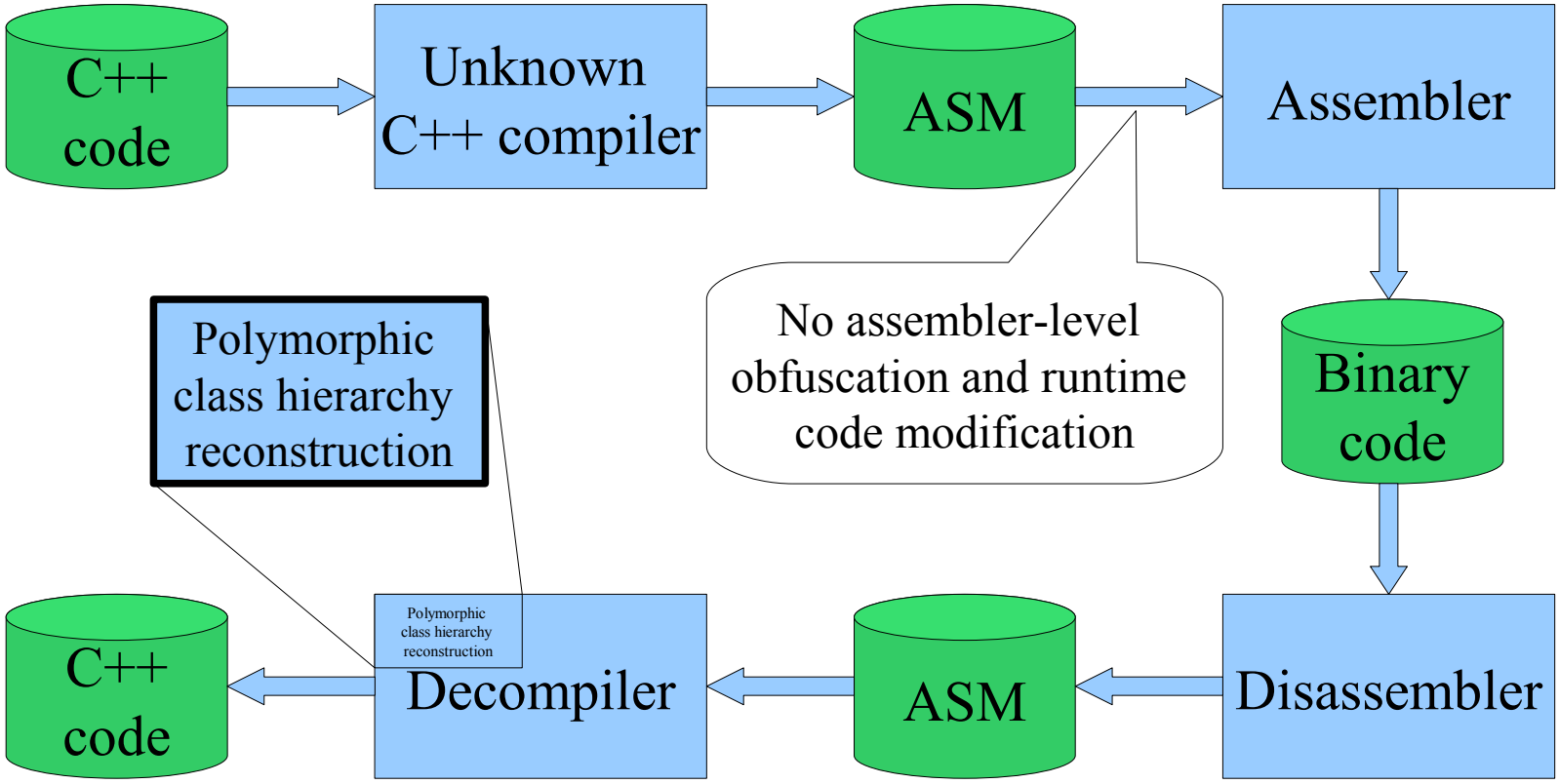
Reconstruction of Class Hierarchies for Decompilation of C++ Programs

Alexander Fokin
Katerina Troshina
Under the supervision of Prof. Alexander V. Chernov

Purpose

- Analysis and maintenance of legacy code.
- Malware analysis.
- Protocol reconstruction.

Problem statement

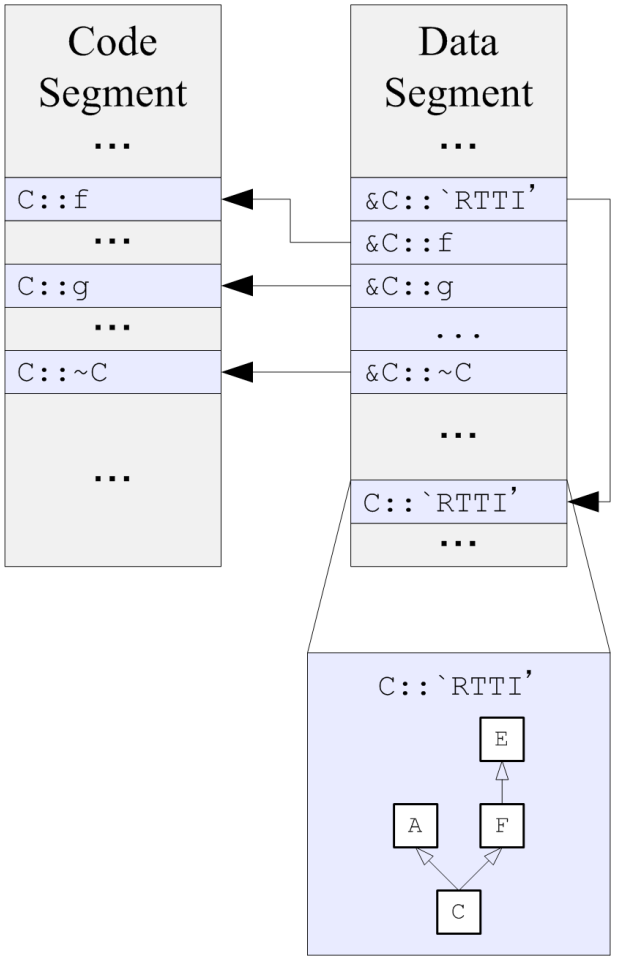


Plan

- Analysis of RTTI structures
- Analysis in case of absence of RTTI structures
  - Reconstructing virtual table inheritance hierarchy
  - Reconstructing polymorphic class hierarchy

Analysis of RTTI structures

- Locate RTTI structures
  - Pointer to RTTI structure precedes the vtable.
  - Vtables are arrays of pointers to functions.
  - Only the first element of vtable is referenced from program code.
- Analyze RTTI structures



Reconstructing virtual table inheritance hierarchy

- Vtable localization.
- Collecting of information on vtable inheritance hierarchy using several rules.

Example rule. *If the size of vtable* B *is less then the size of vtable* D*, then* B *cannot inherit from* D.

- Processing of the collected information to construct a set of vtable inheritance trees.

Reconstructing polymorphic class hierarchy

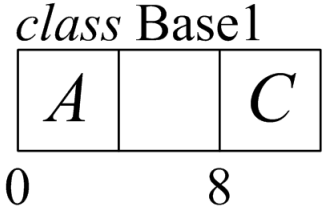
- Interprocedural data flow analysis is used to detect code sites where pointers to vtables are written into memory locations differing by constant offset.

Example

```
mov [esi], offset A::'vtable'
; ...
mov [esi+8], offset C::'vtable'
; ...
mov [esi], offset B::'vtable'
```

Reconstructing polymorphic class hierarchy

- Each such vtable access site is associated with a set of (offset, vtable sequence) pairs.
- Each such set determines a class.



Example

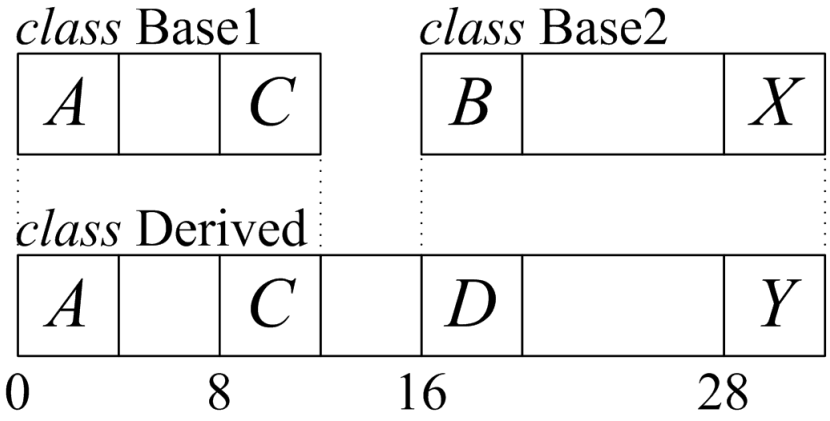
```
mov [esi], offset A::'vtable'
; ...
mov [esi+8], offset C::'vtable'
; ...
mov [esi], offset B::'vtable'
```

Set of (offset, vtable sequence) pairs:

(0, [*A*, *B*]), (4, [*C*])

Reconstructing polymorphic class hierarchy

- Class inheritance relation inference is performed. It uses previosly reconstructed vtable inheritance trees.



Here *B* is a direct base of *D* and *X* is a direct base of *Y*.

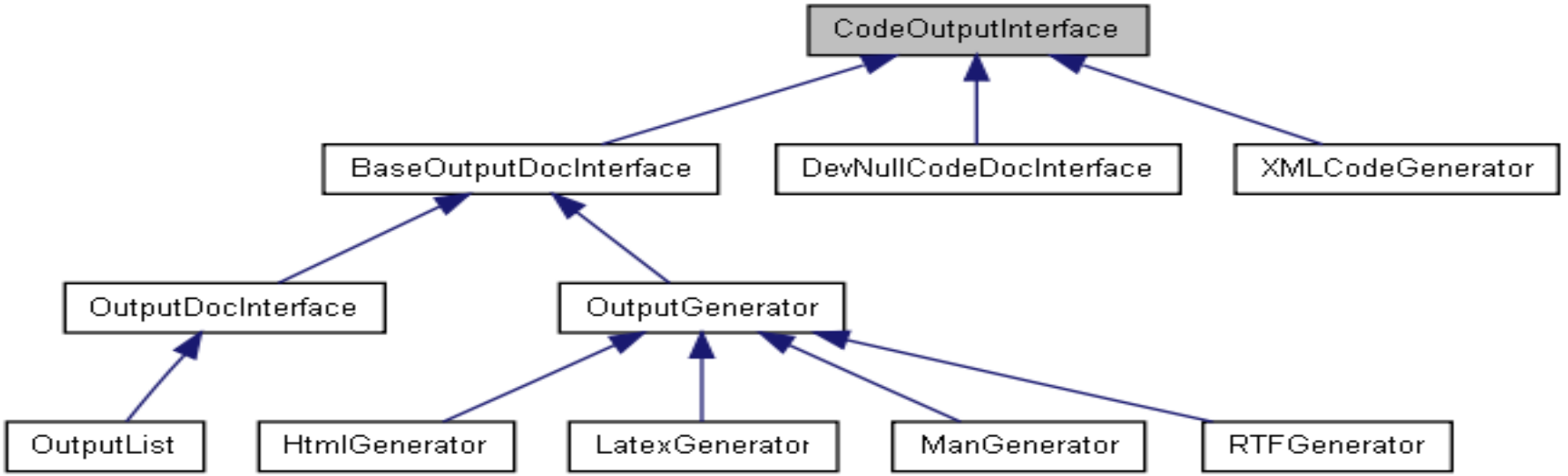
Test results

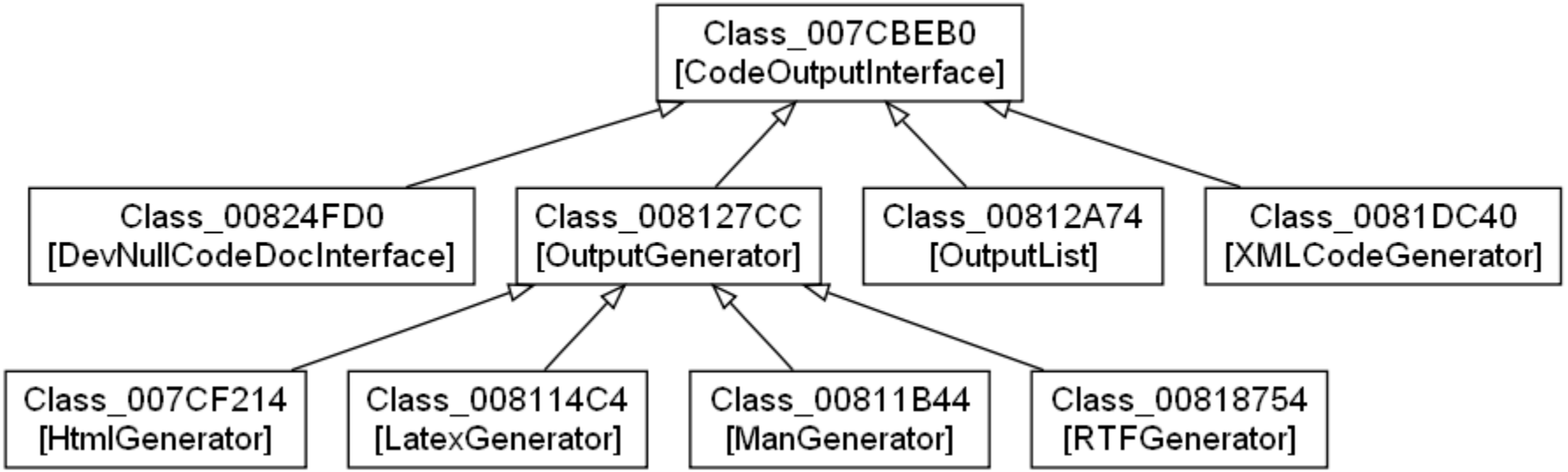
Application	doxygen	shareaza	notepad++
Vtables found	415	1128	95
Vtable mismatches	8.6%	4.1%	4.0%
Classes found	401	1108	95
Non-classes	0.9%	0.6%	0.0%
Class mismatches	9.7%	6.5%	4.0%

Results

- Described approach was implemented in a plugin for IDA Pro interactive disassembler and is available for download at `http://decompilation.info`.

Questions?


CodeOutputInterface
BaseOutputDocInterface
DevNullCodeDocInterface
XMLCodeGenerator
OutputDocInterface
OutputGenerator
OutputList
HtmlGenerator
LatexGenerator
ManGenerator
RTFGenerator


Class_007CBEB0
[CodeOutputInterface]
Class_00824FD0
[DevNullCodeDocInterface]
Class_008127CC
[OutputGenerator]
Class_00812A74
[OutputList]
Class_0081DC40
[XMLCodeGenerator]
Class_007CF214
[HtmlGenerator]
Class_008114C4
[LatexGenerator]
Class_00811B44
[ManGenerator]
Class_00818754
[RTFGenerator]