

Exchange formats: Some problems, a few results, and a cool name

Michael Godfrey
Ivan Bowman
and others ...
University of Waterloo



CSER / CASCON 1999

Exchange Formats

- What?
- Why?
- How?
- Whose?
- Problems?
- Volunteers?

November 7, 1999

CSER / CASCON 1999

2

References

- "Connecting architecture reconstruction frameworks", by Bowman, Godfrey, and Holt.
 - *Proc. of CoSET '99*, to appear in *Journal of Information and Software Technology*.
- "An architecture for interoperable program understanding tools" (CORUM), by Woods et al.
 - *Proc. of IWPC '98*
- "CORUM II", by Kazman, Woods, and Carrière.
 - *Proc. of WCRE'98*.

November 7, 1999

CSER / CASCON 1999

3

What?

- CASCON '98: CSER members identified opportunities for re-use between tools
- Want to be able to map software "facts" extracted by different tools to a common format.
- Want different levels of abstraction supported (code, architecture, etc.)

November 7, 1999

CSER / CASCON 1999

4

Why?

- Different strengths, bugs, detail level, robustness, languages supported, ...
 - acacia, cfx, Datrix, Rigi, Dali
- Research cross fertilization, validation
- Plug 'n play subtools (esp. new uses)
 - extractor, reasoning engine, clusterer, visualizer
- Commercial linkage

November 7, 1999

CSER / CASCON 1999

5

My Selfish Reason

- Want to opportunistically steal tools for use in the BEAGLE system
 - BEAGLE models *evolution* of software systems over time.
 - Need extractors, fact manipulators, visualizers, etc.
 - Dealing with scale, incrementality, flexible middle are key issues.

November 7, 1999

CSER / CASCON 1999

6

Exchange Format Requirements

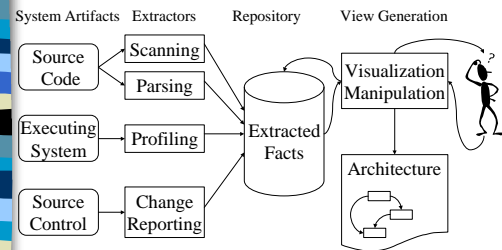
- Support multiple source languages
- Scale to large systems (e.g., 10 MLOC)
- Provide mapping to source code
- Support static & dynamic dependencies
- Incremental approach
- Must be extensible, allowing new schemes to be defined as needed

November 7, 1999

CSER / CASCON 1999

7

Architectural Reconstruction



November 7, 1999

CSER / CASCON 1999

8

TAXForm –TA Exchange Format

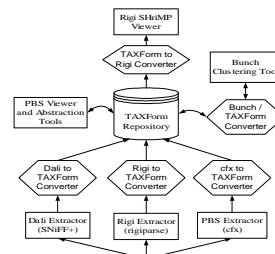
- Idea: provide a common format and *converters* to allow tools to interoperate
- Two parts to an exchange format:
 - Syntax of data (representation in files)
 - Semantic structure (schemas)
- We chose TA syntax (others are attractive)
- Tool developers may define their own schemas as needed

November 7, 1999

CSER / CASCON 1999

9

TAXForm Utopia

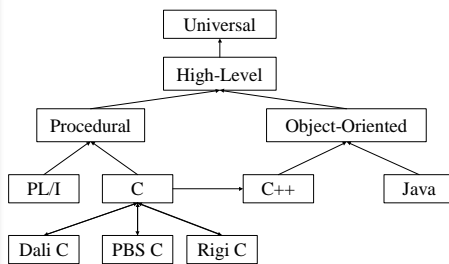


November 7, 1999

CSER / CASCON 1999

10

Transforming Between Schemas

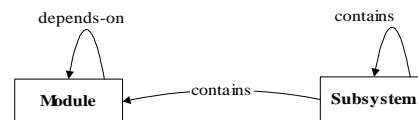


November 7, 1999

CSER / CASCON 1999

11

TAXform — High level schema

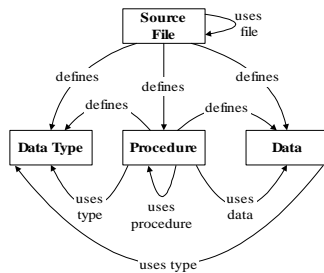


November 7, 1999

CSER / CASCON 1999

12

TAXform — Procedural schema



November 7, 1999

CSER / CASCON 1999

13

Problems

- Different extractors use different:
 - syntax (and storage formats)
 - semantic models (schemas)

November 7, 1999

CSER / CASCON 1999

14

Problem: Naming

- Each entity must have unique ID
- Source languages may allow two code elements to have the same name
 - `typedef int T;`
 - `struct T { ... };`
- To combine facts, we need a common naming scheme
- Ivan has a Java scheme; C/C++?

November 7, 1999

CSER / CASCON 1999

15

Problem: Line Numbers

- We require a mechanism to get from an entity back to source code
- An obvious solution: `file + line#`
 - Want same file name on different machines
 - Some entities are defined on a range of lines, or non-contiguous ranges of lines (e.g., namespaces)

November 7, 1999

CSER / CASCON 1999

16

Problem: Resolution

- For each reference in source code, we can determine the reference *target*
- Several resolution strategies are used:
 - *No resolution* (each reference is an entity)
 - *Resolved to declaration* (in a header file)
 - *Resolved to static definition* (entity body)
 - *Resolved to dynamic definition* (virtual functions, pointers)

November 7, 1999

CSER / CASCON 1999

17

Some dry runs

- `rigi2pbs`, `acacia2pbs*` (C++) [Bowman]
- `dali2pbs*` [Carrière]
- `cia2rigi` [KAC]
- `cia2pbs`, `acacia2pbs` (C) [Godfrey]
- `acacia2pbs` (C++) [Lee, Fung]

* special purpose use

November 7, 1999

CSER / CASCON 1999

18

Some experiments [Bowman]

System	Size (KLOC)	Language	Extractor
Jikes	77	C++	Acacia
Linux	800	C	Dali, cfx
Mozilla	904	C	Rigi
Nachos	10	C++	Acacia

November 7, 1999

CSER / CASCON 1999

19

acacia2pbs — An Experiment

- My immediate goal:
 - want to be able to use CIA/acacia extractor as plug-in replacement for *cfx* within PBS (i.e., generate *factbase.rsfl*)
 - *cfx* gets some facts wrong, doesn't extract enough detail for *arch. repair* [Tran]
- Also, get some experience for BEAGLE

November 7, 1999

CSER / CASCON 1999

20

acacia2pbs — Nuts and bolts

- Acacia extractor similar to *cfx*:


```
Ccia -D<arg> -I<arg> *.c
```

 generates *entity.db*, *relationship.db*
- Use SQL-like queries to get raw text output:


```
cdef -u func - def=dec
cdef -u - - m -
```

 produces “,” delimited textual output

November 7, 1999

CSER / CASCON 1999

21

acacia2pbs — Nuts and bolts

- Pretty much 1:1 (1:n) relationship with *factbase.rsfl* output via *awk*
- ... but “linkcall” harder as
 - acacia already does resolution of “f calls g” to the function defs;
 - *cfx* does resolution at a later stage
 - no transitive closure for “includes”
- Solution: simple *grok* program

November 7, 1999

CSER / CASCON 1999

22

acacia2pbs — Nuts and bolts

- Unique IDs and fake polymorphism:
 - May be multiple function defs named “f”
 - How to disambiguate?
- PBS just assumes it won't happen.
- Acacia uses hashing to unique IDs, but not clear what it does on collisions.
- I use “foo.c#f” as entity name, demangle at end of translation.

November 7, 1999

CSER / CASCON 1999

23

acacia2pbs — Summary

- Works well; adds more detail than *cfx*; acacia *factbase* slightly more accurate
- Example: *ctags-3.0* (10 KLOC, 5000 facts)
 - *cfx/fbgen*: 12 seconds to create *factbase.rsfl* on fast Sparc
 - *acacia2pbs*: 9 seconds to create acacia database + 30 seconds for my naïve scripts to convert it to *factbase.rsfl*

November 7, 1999

CSER / CASCON 1999

24



Volunteers?

- What real interest is there?
It *sounds* like a good idea ...
- How / why will your group use a common exchange format?
- Lots of talk, some (mostly isolated) action...
- “Good enough” good enough?

November 7, 1999

CSER / CASCON 1999

25