

# Defining, Transforming, and Exchanging High-Level Schemas

*A guided journey through the outback*



Presented by Michael W. Godfrey  
 Software Architecture Group (SWAG)  
 Dept of Comp Sci, Univ of Waterloo

This presentation is available from  
<http://plg.uwaterloo.ca/~migod/papers/>

## What is a High-Level Schema?

My answer:

*Any schema above the statement level*

I see two distinct levels of abstraction:

1. Programming language entity level
  - Entities are (shared) fcn, vars, types, classes, ...
2. Architectural level
  - Entities are modules, subsystems, classes, interfaces, ...

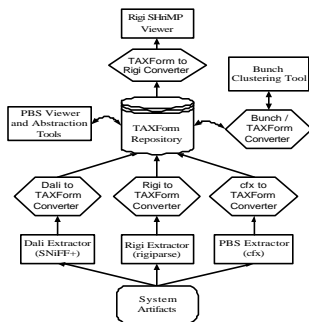
## Previous Work

- Lots of
  - motivational work
  - *ad hoc* extractor snarfing
  - experimental translation mechanisms
- Examples (many others exist)
  - CORUM I and II
  - GRAX
  - TAXForm (TA eXchange FORMat) using Acacia, Rigi, parse
  - Rigi using VisualAge C++
  - Dali using Sniff+

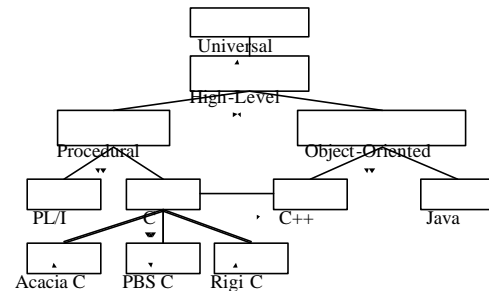
## My (selfish) goals

- I would like to be able to use other extractors ...
  - Want to perform architectural analyses of systems written in languages other than C
  - Want to implement *BEAGLE*  
 (a tool for exploring software evolution)
- ... but extractors differ in languages modelled, level of detail, robustness, bugs, data format, ...
  - I want to be able to convert data between tools.
  - Need agreement (awareness) from tool creators

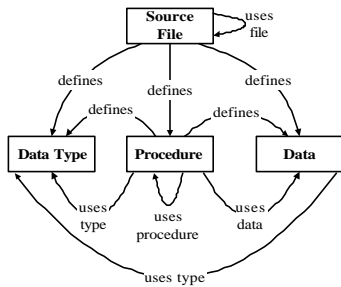
## TAXForm Utopia



## Transforming Between Schemas



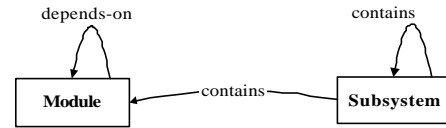
## TAXForm — Procedural schema



WCRE 00 -- High Level Schemas

7

## TAXForm — High level schema



WCRE 00 -- High Level Schemas

8

## Back to my (selfish) goals

- Would like to concentrate on procedural and OO languages.
  - Others are interested in COBOL, JCL *etc.*
- I am interested in *high-level* info (f calls g)
  - but not in ASGs, code-level metrics
- Need to agree on
  - Syntax
  - Level of granularity and detail
  - What to do in case of X *e.g.*, X = “missing files”

WCRE 00 -- High Level Schemas

9

## My schema wish list

*[influenced by Acacia's C and C++ data models]*

### Top-level programming language entities:

- functions, variables, constants, type definitions (procedural languages)
- methods, class member data, static methods and member data (object-oriented languages)

### Entity containers:

- files, modules, classes, packages

WCRE 00 -- High Level Schemas

10

## My schema wish list

### Entity attributes:

- Name, unique identifier (UID -- see next section)
- UID of container, UID of containing file (if container is not a file)
- Signature/data type
- Line number information (see below)
- Declared scope/visibility, static or not, final or not
- Definition or declaration (see below)

### Entity container attributes:

- name, UID
- relative path (if a file)
- version identifier (if provided)
- UID of container (if not a file), UID of containing file (if not a file)

WCRE 00 -- High Level Schemas

11

## My schema wish list

### Relationships:

- Function calls, variable uses
- Line number information (see below)
- Container use/inclusion (by other containers)
- Inheritance (various kinds)
- “Friendship”, various template relationships

### Relationship attributes:

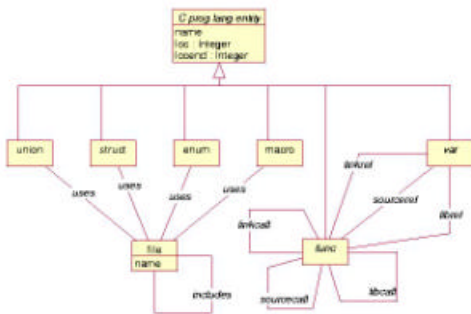
- Line number information (see below)
- Scope/permission of inheritance

WCRE 00 -- High Level Schemas

12



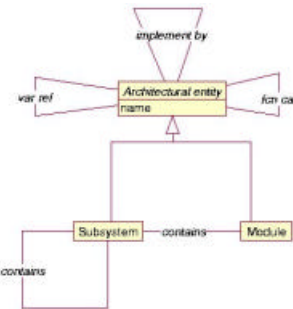
## PBS C Language E/R View



WCRE 00 – High Level Schemas

19

## PBS Architectural Schema



WCRE 00 – High Level Schemas

20

## Acacia [Chen, Gansner *et al.* @ AT&T]

- History:
  - CIA → CIAO → Acacia
- Consists of
  - C and C++ extractors
  - SQL-like query engine
  - visualization with auto-layout

WCRE 00 – High Level Schemas

21

## Acacia C++/C Schemas

- Entity attributes:
  - Hex UID, name, kind (file, function, type, var, macro), filename, datatype (string), typeclass (enum, struct, etc.), linenum info for def/dec, def/dec/undef, param list, template info, scope, storage spec (static, const, inline, inline virtual, etc.), signature
- Relationship attributes:
  - Linenum info, rel. kind (refers, contains, inherits, instantiates, typedef, etc.), relationship scope

WCRE 00 – High Level Schemas

22

## Acacia Queries

- SQL-like queries for entities and relationships produces “;” delimited textual output:

```
% ksh cdef -u fu closeTagFile
26f53ece:closeTagFile:function:entry.h:void:regular:83:0:83:
dec:00000000:(const boolean)::extern;;;
76e7ae31:closeTagFile:function:entry.c:void:regular:551:553:
563:def:00000000:(const boolean)::extern;;;

% ksh cref -u - - m - file2='osdeps.h'
<all entity1 attrs> ; <all entity2 attrs > ; <rel attrs>
```

WCRE 00 – High Level Schemas

23

## ctags, cxref, cscope

- These are “open source” Unix tools that perform extractions:
  - ctags extracts only entity info
    - e.g., file, name, line num, kind, etc
    - works with C, C++, Eiffel, Fortran, and Java.
    - Used for fast context switching while editing source code with vim/emacs
  - cxref generates cross-reference table for C systems.
    - Often used for webifying source code (e.g., Linux, Mozilla).
  - cscope used for program comprehension of C systems (e.g., who calls f, who uses v)
    - Older commercial Unix tool, recently open sourced.

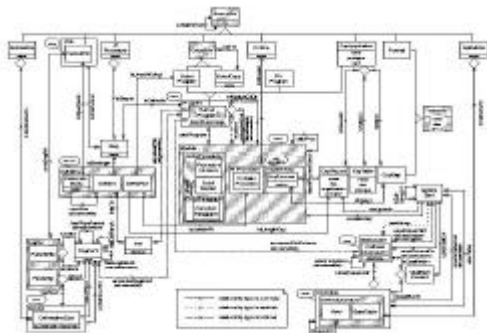
WCRE 00 – High Level Schemas

24





## GUPRO Multi-Language Model



WCRE 00 – High Level Schemas

37

## Summary — High-Level Schemas

- Lots of sticky issues at the prog. lang. level:
  - To pre- or not to pre-process
  - Entity resolution often not done (*e.g.*, Datrix)
  - What is a function: def, dec, polymorphism, overloading, templates, ...
  - How to deal with missing libraries, incremental extractions, versioned extractions, non-ANSI-isms, ...
- Conceptual gaps:
  - COBOL/JCL world very different from C/C++/Java world
  - “*I didn't know you wanted full includes info...*”

WCRE 00 – High Level Schemas

38

## Summary — Good News

- Many of us seem to be doing similar kinds of extractions. It seems like that:
  - Many extractors *can* be used within other tools
  - Some form of common interchange format *is* feasible, tho it may not please everyone.
- Challenges:
  - May want to use multiple tools together
    - I have been working on a standalone `cxref`-based hack to add full `includes` information to a BAUHAUS converter
  - Can we take advantage of the web to set up some sort of distributed fact extraction/conversion factory? [Holt]

Q: Are you game?

WCRE 00 – High Level Schemas

39