Towards a Standard Schema for C/C++

Rudolf Ferenc¹
Susan Elliott Sim²
Richard C. Holt³
Rainer Koschke⁴
Tibor Gyimóthy¹

¹ University of Szeged  ³ University of Waterloo
² University of Toronto  ⁴ University of Stuttgart
Introduction

- GXL (Graph eXchange Language)
- Schema
- Model
- Standard Schema
- Catalogue of issues to be considered for defining a standard schema
Motivation

- Building tools is expensive
- A SEF offers an infrastructure for tool interoperability and reuse

Fundamental issues:
- Standard schema for C/C++ at AST level
- Defining a schema rather than a front end
- Independence from parsing technologies
- ANSI C++
Classifications of existing schemas

- According to the methods for deriving the schema from the grammar:
  - Automatically derived schemas
  - Manually derived schemas

- According to language dependence
  - Language-specific
  - Language-independent

- API
Datrix Schema

- Bell Canada
  - Represents ASTs for C/C++ and Java
  - Common front end for software analysis and assessment

- University of Waterloo
  - Andrew Malton, Tom Dean, Ric Holt
  - CPPX – open source, based on GNU GCC
  - Freely available
Columbus Schema

- University of Szeged & Nokia Research Center
  - C/C++ Schema
  - Columbus front end for various re- and reverse engineering tasks
- FrontEndART Ltd.
  - New owner of Columbus
  - Freely available soon for academic purposes
Catalogue of representation Issues

- Lexical Structure
  - Preprocessing
  - Line/column number problem
  - Project handling problem

- Syntax
  - Templates
  - Types
  - Functions
  - Statements

- Semantics
  - Naming problem
  - Resolution problem
Datrix representation of types

```cpp
template <class T, int Size>
class Array{
    T arr[Size];
public:
    virtual const T& get(int idx) {
        T &t = arr[idx];
        return t;
    }
};
```
Columbus representation of types

```
template <class T, int Size>
class Array{
T arr[Size];
public:
    virtual const T& get(int idx) {
        T& t = arr[idx];
        return t;
    }
};
```

The type representation is similar to the type of function "get".
Discussion

- Comparison of the two schemas
- Other approaches
Conclusion

- Continue our work on creating a standard schema for C++ at the AST level
- A hope to advance the state of tool interoperability
- Mailing list:
  - http://rgai.inf.u-szeged.hu/mailman/listinfo/gxl-cpp