Infrastructure to Explore, Analyze, and Manipulate Source Code

www.srcML.org

NSF CNS 20-16465/16452 and CNS 13-05292/05217

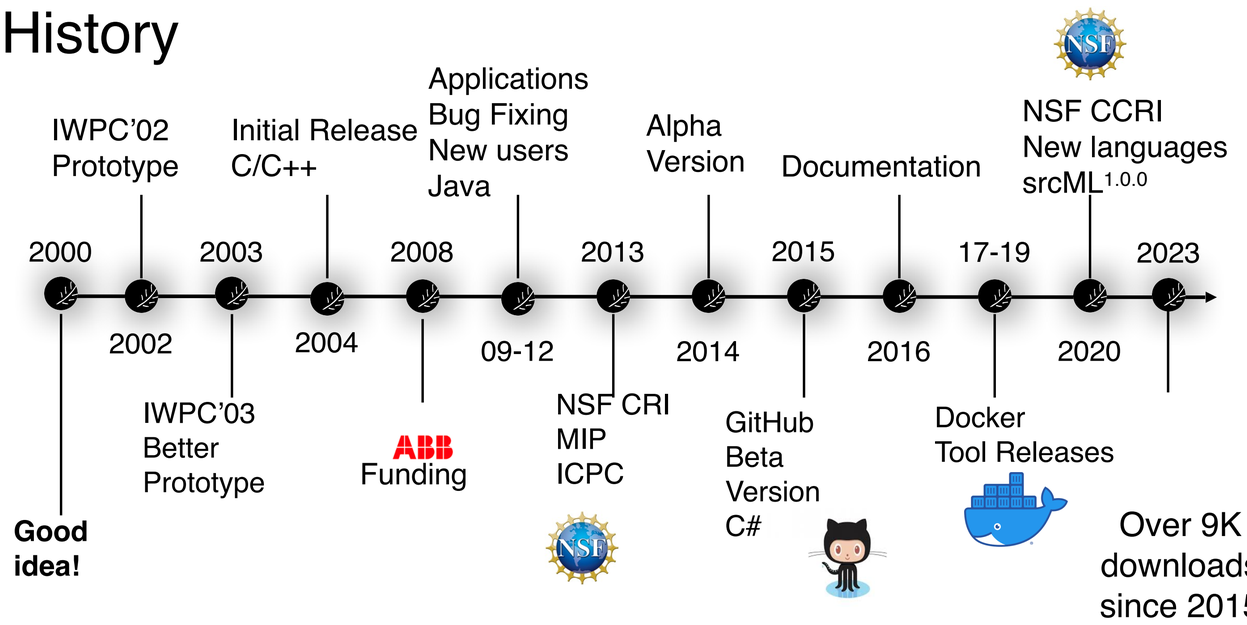
Jonathan I. Maletic
jmaletic@kent.edu
Department of Computer Science
Kent State University
Ohio, USA

Michael L. Collard
collard@uakron.edu
Department of Computer Science
The University of Akron
Ohio, USA

srcML (sõrs em el), *n*. **1.** an infrastructure for the exploration, analysis, and manipulation of source code. **2.** an XML format for source code. **3.** a lightweight, highly scalable, robust, multi-language parsing tool to convert source code into srcML. **4.** an open source software application licensed under GPL.

GPL

History



Tools

Tools provided and custom built are used to query, extract data, and transform source code.

Models

External models of the code such as PDG, UML, call graphs can be built in XML

XML

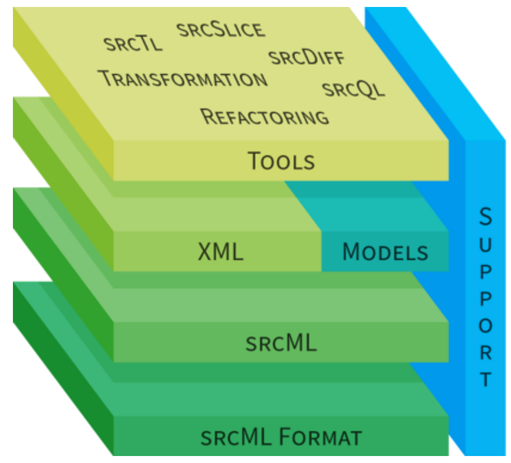
The full range of XML technologies can be applied to the srcML format.

srcML

The srcml CLI is used to convert entire projects from and to source code and the srcML format. Languages supported include C, C++, Java, and C#.

srcML Format

The srcML format represents source code with all original information intact, including whitespace, comments, and preprocessing statements.



Support

A multi-university team currently supports the infrastructure.

srcML Enabled Research

- Static analysis: slicing, pointer analysis, PDG, etc.
- Fact extraction, custom profiling
- Computing metrics
- Refactoring, transformation
- Syntactic differencing
- Reverse engineering UML class diagrams,
- Method/class stereotyping
- Preprocessor analysis
- Identification of security vulnerabilities in code

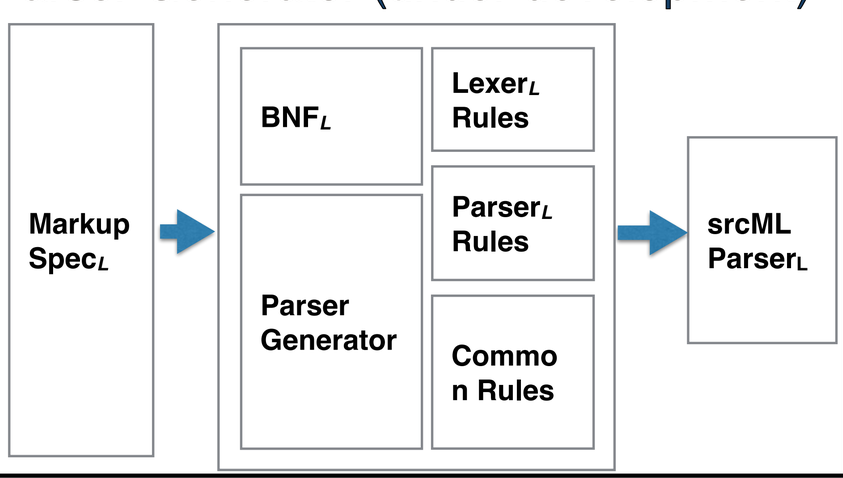
Language Support

- C11, K&R C
- C++17, Qt extension
- Java SE8
- C# standard ECMA-334
- OpenMP

Coming soon:

- Python, Swift, JavaScript
- Rust? Go? Ruby?

Parser Generator (under development)

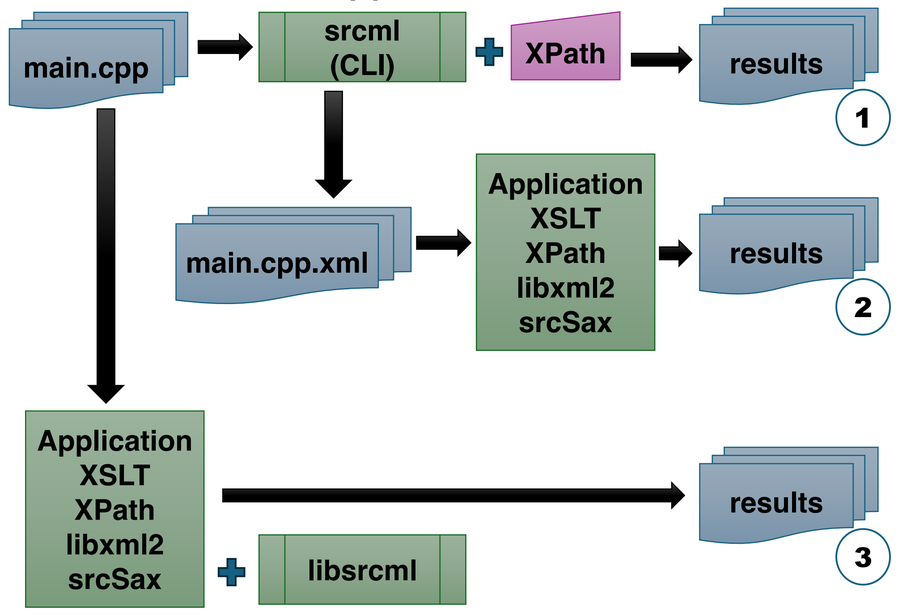


srcML Embeds Abstract Syntax Information into the Code

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<unit xmlns="http://www.srcML.org/srcML/src" xmlns:cpp="http://www.srcML.org/srcML/cpp"
      revision="1.0.0" language="C++">
<cpp:include>#<cpp:directive>include</cpp:directive> <cpp:file>"rotate.h"</cpp:file></cpp:include>

<comment type="line">// rotate three values</comment>
<function><type><name>void</name></type> <name>rotate</name>
   <parameter_list>(
         <parameter><decl><type><name>int</name><modifier>&amp;</modifier></type> <name>n1</name></decl></parameter>,
         <parameter><decl><type><name>int</name><modifier>&amp;</modifier></type> <name>n2</name></decl></parameter>,
         <parameter><decl><type><name>int</name><modifier>&amp;</modifier></type> <name>n3</name></decl></parameter>
     )</parameter_list> <block>{<block_content>

  <comment type="line">// copy original values</comment>
  <decl_stmt><decl><type><name>int</name></type>
                         <name>tn1</name> <init>= <expr><name>n1</name></expr></init></decl>;</decl_stmt>
  <decl_stmt><decl><type><name>int</name></type>
                         <name>tn2</name> <init>= <expr><name>n2</name></expr></init></decl>;</decl_stmt>
  <decl_stmt><decl><type><name>int</name></type>
                         <name>tn3</name> <init>= <expr><name>n3</name></expr></init></decl>;</decl_stmt>

  <comment type="line">// move</comment>
  <expr_stmt><expr><name>n1</name> <operator>=</operator> <name>tn3</name></expr>;</expr_stmt>
  <expr_stmt><expr><name>n2</name> <operator>=</operator> <name>tn1</name></expr>;</expr_stmt>
  <expr_stmt><expr><name>n3</name> <operator>=</operator> <name>tn2</name></expr>;</expr_stmt>
</block_content></block>}</function>
</unit>
```

Using srcML

- srcML command line client + XML tools
- libsrcml C API + Application code



Tools Built with srcML (released)

- srcSAX – a sax framework to use srcML
- srcSlice – scalable forward static slicer
- srcPtr – lightweight pointer analysis
- srcUML – Source to UML class diagram
- stereoCode – method/class stereotypes
- nameCollector – get all identifier names

Under Development

- srcQL – syntactic aware query language
- WASM playground
- srcTL – transformation language
- srcDiff – syntactic differencing (CIRC award)

Implementation

- Parsing technology in C++ with ANTLR
- Uses libxml2, libarchive, CLI11
- No loss of original source code
- Markup of preprocessor, templates, etc.
- Current speed: 262 KLOC/sec
- srcML to text: ~3.4 (~1.5 compressed)
- Linux Kernel: 2 minutes
- Allows for various input sources: files, directories, tar.gz, etc.

```
srcml  linux-6.6.tar.xz  -o linux-6.6.xml.gz
```

Release and Deployment

- srcML.org
- github.com/srcML
- Testing – over 50K individual parser tests + stress testing
- Documentation, tutorials
- GitHub - issue tracking and dev Executables (multiple versions):
  - Windows
  - macOS
  - Ubuntu, Fedora
  - CentOS
  - openSUSE