Using Static Program Analysis to Automatically Document Source Code Classes and Methods with Stereotypes

www.srcML.org

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

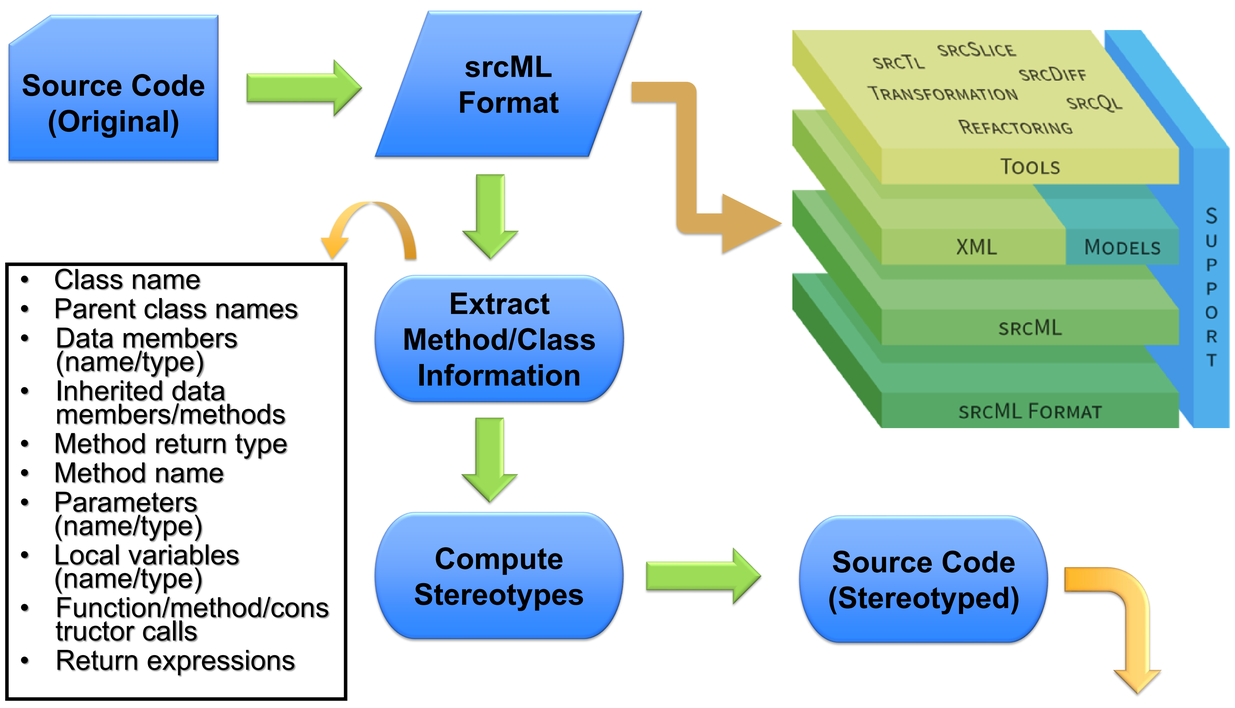
Ali F. Al-Ramadan
aalramad@kent.edu
Department of Computer Science

Advisor:
Jonathan I. Maletic
jmaletic@kent.edu
Department of Computer Science

KENT STATE UNIVERSITY

Stereocode. 1. a comprehensive tool for automatic method and class stereotype identification. 2. a multi-language approach for understanding and documenting the design of software systems through static stereotype analysis. 3. a tool that enables researchers and developers to explore and analyze source code at a higher level of abstraction

STEREOCODE IMPLEMENTATION



Class Stereotype
entity
minimal-entity
data-provider
command
boundary
factory
control
pure-control
large-class
lazy-class
degenerate
data-class
small-class
empty

Category	Method Stereotype
Structural Accessors	get
	predicate
	property
	accessor
Structural Mutators	set
	command
	non-void-command
Creational	constructor
	copy-constructor
	destructor
	factory
Collaborational	collaborator
	controller
Degenerate	stateless
	wrapper
	empty

C++

Example from HippoDraw, a statistical data analysis tool

```
// @Stereotype data-provider boundary
class MDL_HIPPOPLOT_API PeriodicBinaryTransform :
public BinaryTransform
{
public:
  // @Stereotype get
  double xOffset() const;

  // @Stereotype set
  void setYOffset(double y_offset);

  // @Stereotype property collaborator
  double moduloSubX (double x1, double x2) const;

  // @Stereotype stateless
  virtual void adjustValues (AxisModelBase &
     model, hippodraw::Axes::Type  axes, const
     Range & limit);

  ...
}
```

C#

Example from PowerShell, a powerful command-line shell

```
// @Stereotype boundary degenerate
public class ResumeController : Controller
{
  // @Stereotype command
  public async void NoResume();

  // @Stereotype controller stateless
  private bool TryGetRangeHeader(out
    string rangeHeader);

 // @Stereotype wrapper
  private void SetResumeResponseHeaders();

 // @Stereotype command collaborator
 public async void Index();

  ...
}
```

Java

Example from RxJava, a Java library for reactive programming

```
// @Stereotype boundary
public abstract class DefaultRowSorter<M, I>
extends RowSorter<M>
{
  // @Stereotype predicate
  public boolean isSortable(int column);

  // @Stereotype factory controller stateless
  private SortKey toggle(SortKey key);

  // @Stereotype property
  public int getModelRowCount();

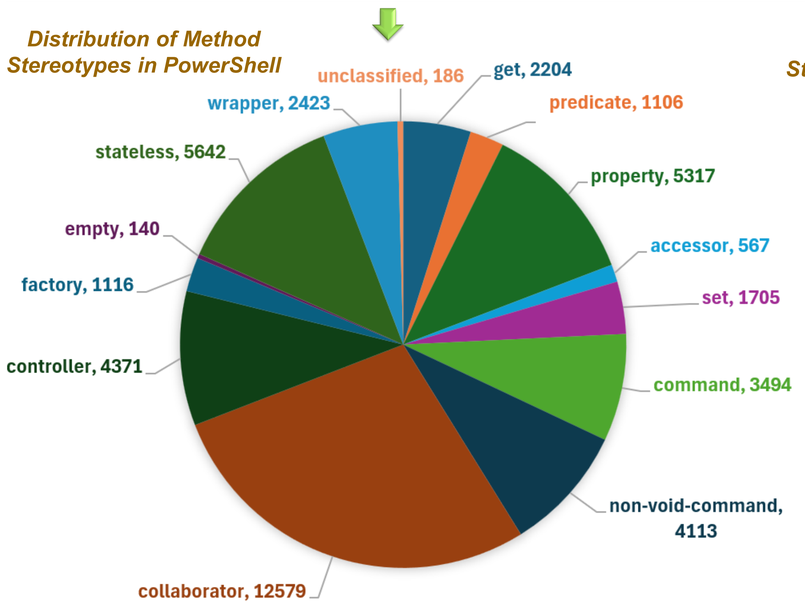
  // @Stereotype get property
  // Possible code smell
  public int convertRowIndexToModel(int index);

  ...
}
```

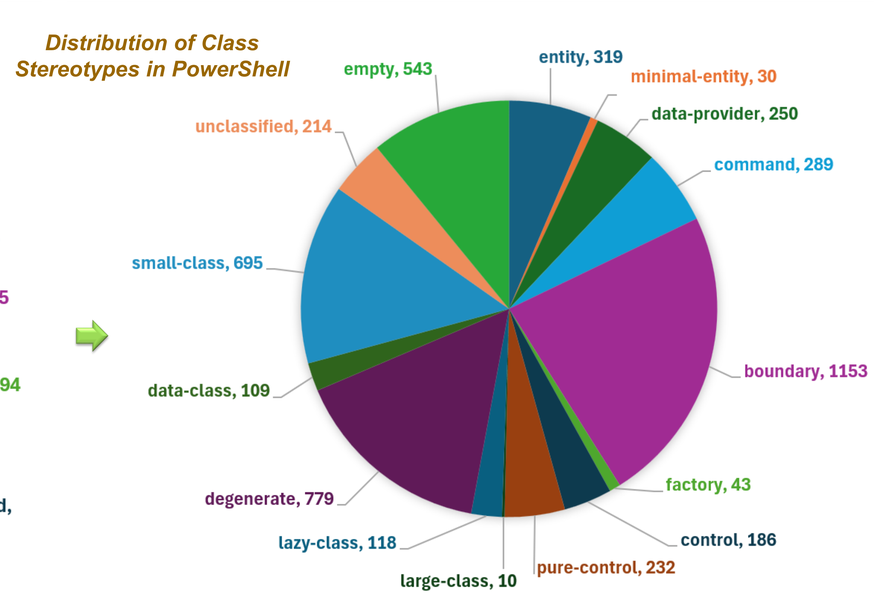
Motivation

- Software systems are becoming complex
- Stereotypes can capture the high-level design and behavior of software systems
- Stereotypes provide a way to quickly grasp the overall architecture of a software system
- Manual documentation of stereotypes is costly

Distribution of Method Stereotypes in PowerShell



Distribution of Class Stereotypes in PowerShell



Features and Language Support

- Supports C++, C#, and Java
- Supports complete software systems as input
- Handles language-specific features: friend functions, generics, structs, interfaces, multi-level inheritance, properties, partial classes
- Multiple output formats: srcML archive, CSV, TXT
- Ability to add/remove/adjust stereotype rules easily

Stereocode Enabled Research

- Create signature descriptors to describe the overall design and architecture of software systems
- Improve feature location in source code
- Generate natural language summaries for methods
- Generate commit messages automatically
- Categorize commits in version control systems
- Study code evolution and change in behavior over time
- Detect code smells

Implementation

- Built using C++ as a command-line tool
- Leverages srcML infrastructure through libsrcml
- Uses XPath queries to extract program information
- Lightweight and fast: ~ 12s to stereotype HippoDraw

```
stereocode hippodraw.xml
```