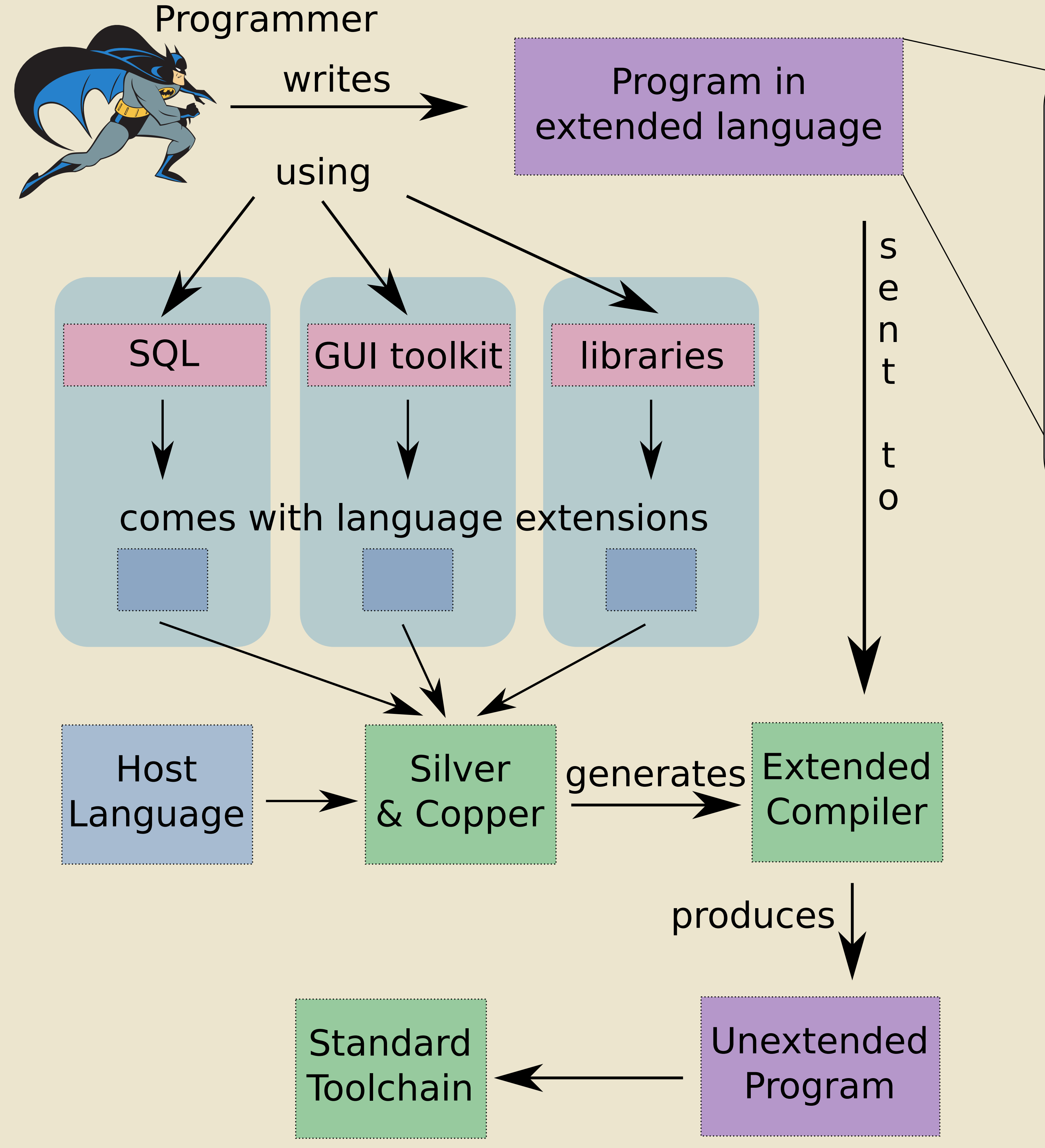
Domain Specific Languages and Language Extension

Minnesota Extensible Language Tools

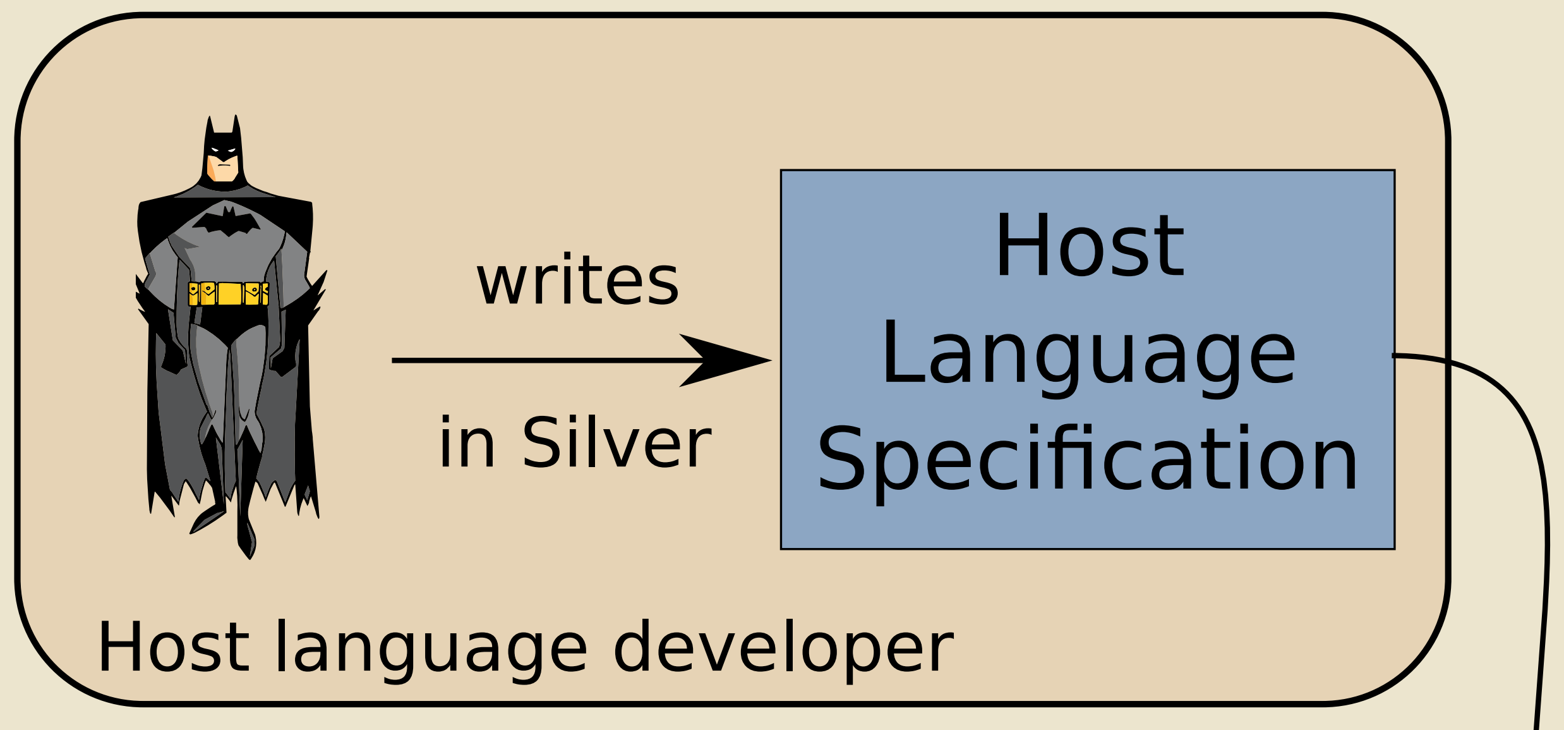
melt.cs.umn.edu

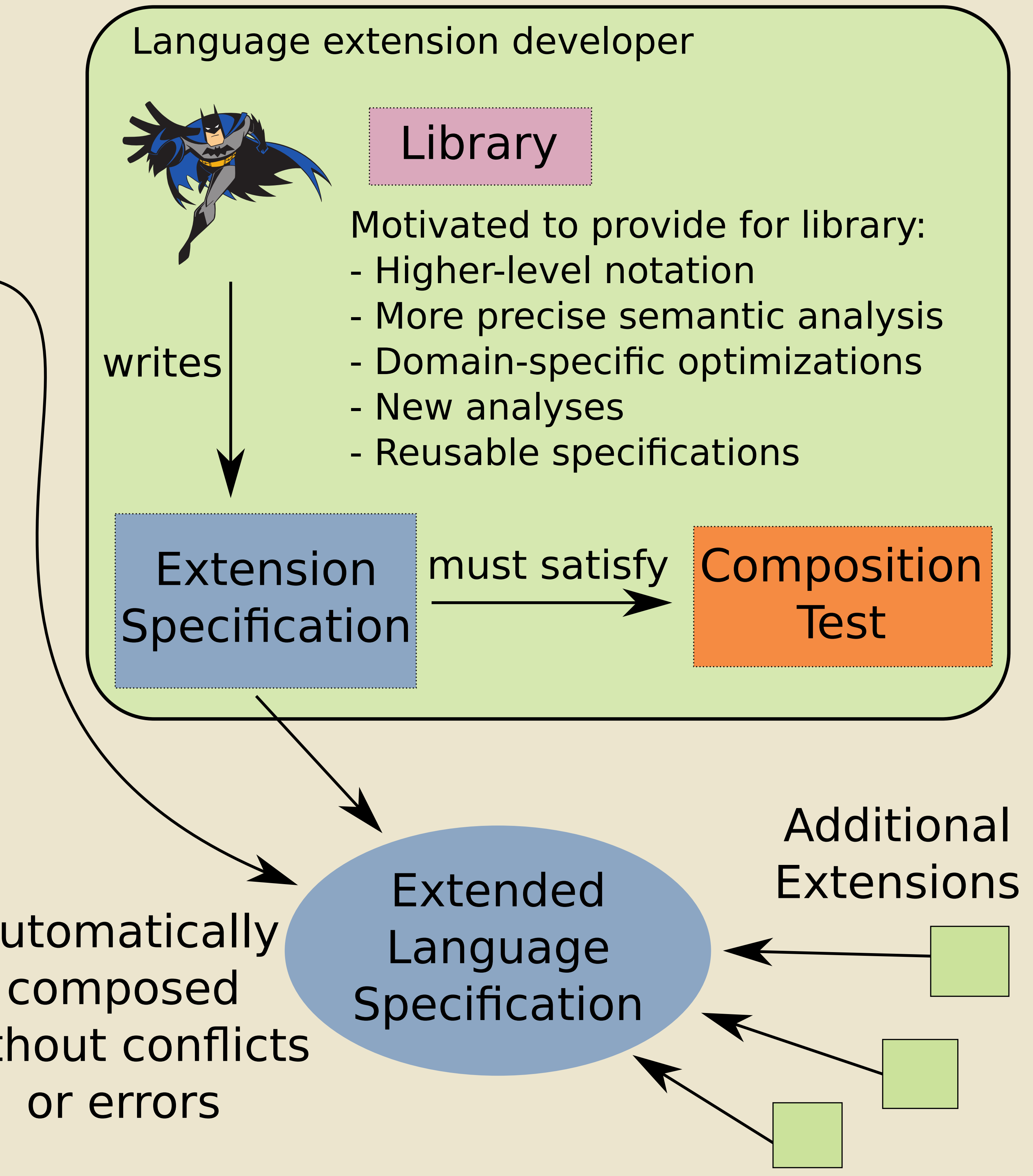


```
class Demo {
  connection db c with table person, location ;

  boolean demo ( ) {
    rs = using db c query {
            SELECT AVERAGE(p.age)
            FROM person AS p, location AS l
            WHERE p.zip = l.zip
            GROUP BY l.state } ;
    ...
  }
}
```

Programmer's perspective





Motivated to provide for library:
- Higher-level notation
- More precise semantic analysis
- Domain-specific optimizations
- New analyses
- Reusable specifications

Automatically composed without conflicts or errors

Domain expert's perspective

Silver

- A language based on attribute grammars
- Used to specify the semantics of host language and extensions
- Modern features: higher-order, reference & collection attributes, powerful types, pattern matching. Not your traditional AGs!
- Forwarding provides for automatic composition of language extensions
- Open source project (LGPL)

Further reading:

Van Wyk, Bodin, Gao, Krishnan. "Silver: an Extensible Attribute Grammar System" (SCP 2010)

Kaminski, Van Wyk. "Integrating attribute grammar and functional programming language features" (SLE 2011)

Van Wyk, de Moor, Backhouse, Kwiatkowski. "Forwarding in Attribute Grammars for Modular Language Design" (CC 2002)

Copper

- A context-aware scanner and deterministic parser generator
- Context makes for a more discriminating scanner, simplifying grammars
- Provides verifiable composition of syntactic extensions to a host language
- Open source project (LGPL)

Further reading:

Schwerdfeger, Van Wyk. "Verifiable Composition of Deterministic Grammars" (PLDI 2009)

Schwerdfeger, Van Wyk. "Context-Aware Scanning for Parsing Extensible Languages" (GPCE 2007)

Schwerdfeger, Van Wyk. "Verifiable Parse Table Composition for Deterministic Parsing" (SLE 2009)

HPC

Language extensions for mining climate data and for sparse matrix computations in materials science

The UMN data mining group is analyzing satellite data to detect the removal of forests. To the left is an illegal mine discovered in Tanzania by their analysis.

Our efforts involve developing language extensions to simplify the development of these analyses, and help scale them up to larger hardware and data sets.

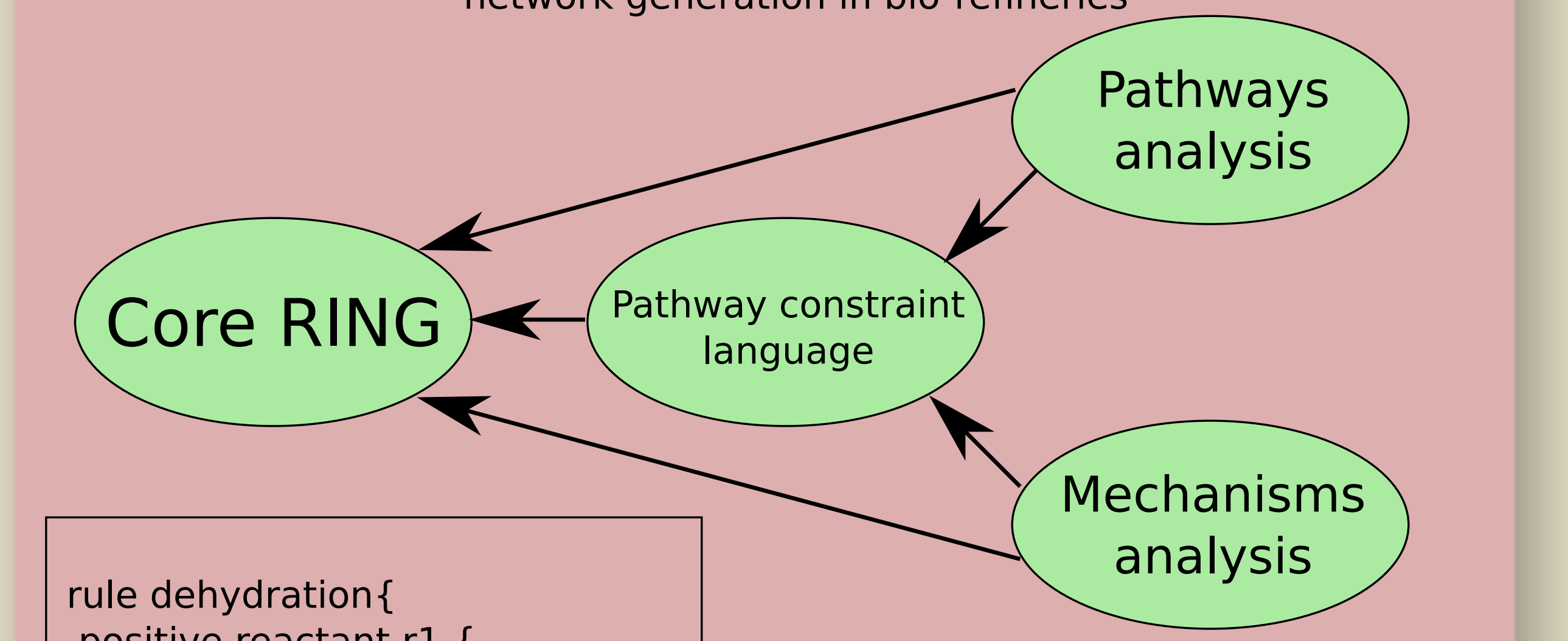
```
// Compute distance between years
Matrix comparison[years][years] (i,j) =
  if i >= j then 0  // Upper triangular
  else sqrt(sum((data[i][:] - data[j][:])^2));
```

From declarative code:
- Generate efficient parallel code
- Make use of stencil libraries
- Make use of Map Reduce frameworks
- Reliably vectorize operations
- Use sparse matrix libraries

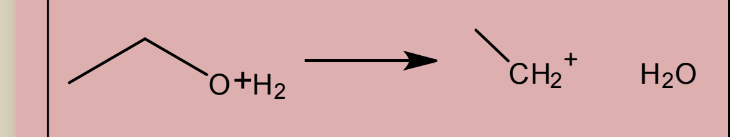
Supported by the NSF Data Intensive Computing and SI2 program

Chemical engineering

A domain-specific language for chemical reaction network generation in bio-refineries



```
rule dehydration{
 positive reactant r1 {
  C labeled c1
  O+ labeled o1 single bond to c1
 }
 break bond (c1, o1)
 modify atomtype (c1, C+)
 modify atomtype (o1, O)
}
```

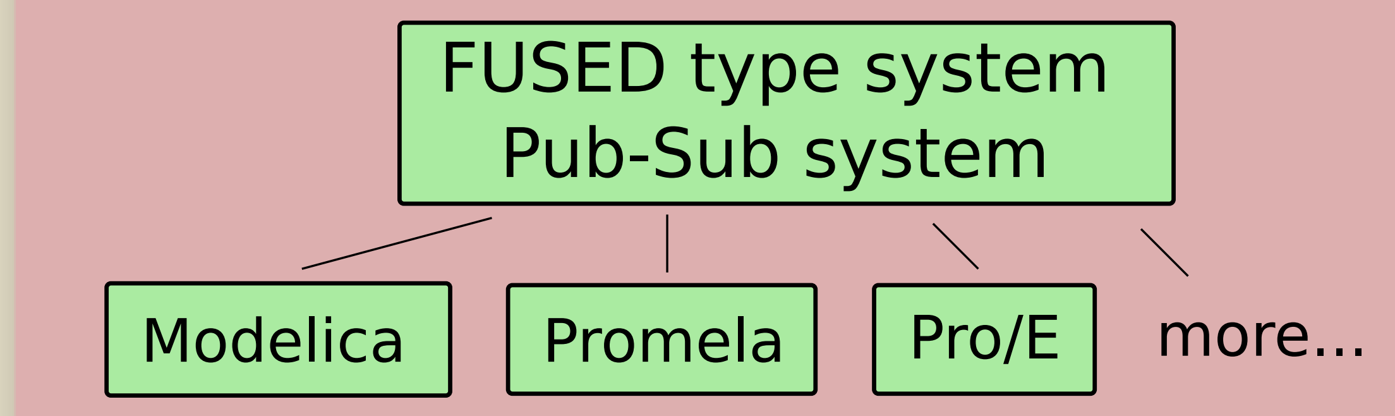


- Feature creep is a common problem with DSLs; avoided by language extension
- Core language purely describes reaction rules
- Made useful by extensions that describe analyses

Supported by the UMN Digital Technology Center DTI Award

Modeling

Modeling languages for embedded systems development



Extend each modeling language:
1) An embedding into the common FUSED type system
2) A syntax for referring to values from other modeling tools
3) A mechanism to translate values to/from the FUSED system

Supported by DARPA META II program and Adventium Labs

Extensible Language Challenges

- Automatically composing concrete syntax specifications.
  Solution: the context-aware scanning and modular composability analysis in Copper.

- Automatically composing specifications of language extension semantics.
  Solution: attribute grammars with forwarding as found in Silver.

- Generating debuggers, interpreters, and IDEs for extended languages
  Solution: additional specifications in object language descriptions and Silver support for generating these tools from them.

Extensible Host Languages

- ableJ: Attribute grammar-based language extesions for Java 1.4
  - extensions include SQL, algebraic types with pattern matching, and boolean expression tables

- ableP: ... for Promela, a modeling language used in embedded systems that supports analysis via model checking.

- ableC: ... for ANSI C. This forms the host langauge for the extensions developed in the high performance computing projects.

People

Eric Van Wyk - evw@cs.umn.edu

Lijesh Krishnan   Ted Kaminski   Kevin Williams

Ming Zhou   Praveen Sugavanam