

Prototalk: an Environment for Teaching, Understanding, Designing and Prototyping OO Languages

Alexandre Bergel, Christophe Dony, Stéphane Ducasse
University of Bern (Switzerland)
LIRMM, Montpellier (France)

bergel@iam.unibe.ch

Outline

1. Prototype OO Languages
2. Expressing Different Languages
3. Prototalk
4. Example: Newtonscript
5. Conclusion

Prototype OO Languages

- Classless paradigm
- A prototype contains its state and behavior
- Link between prototypes is usually done using:
 - Delegation
 - Composition

Different Languages

- Self: Unification of variables and methods and multiple parents
- ObjectLisp: variables distinct from methods
- NewtonScript: two kinds of delegation over two links

Prototalk

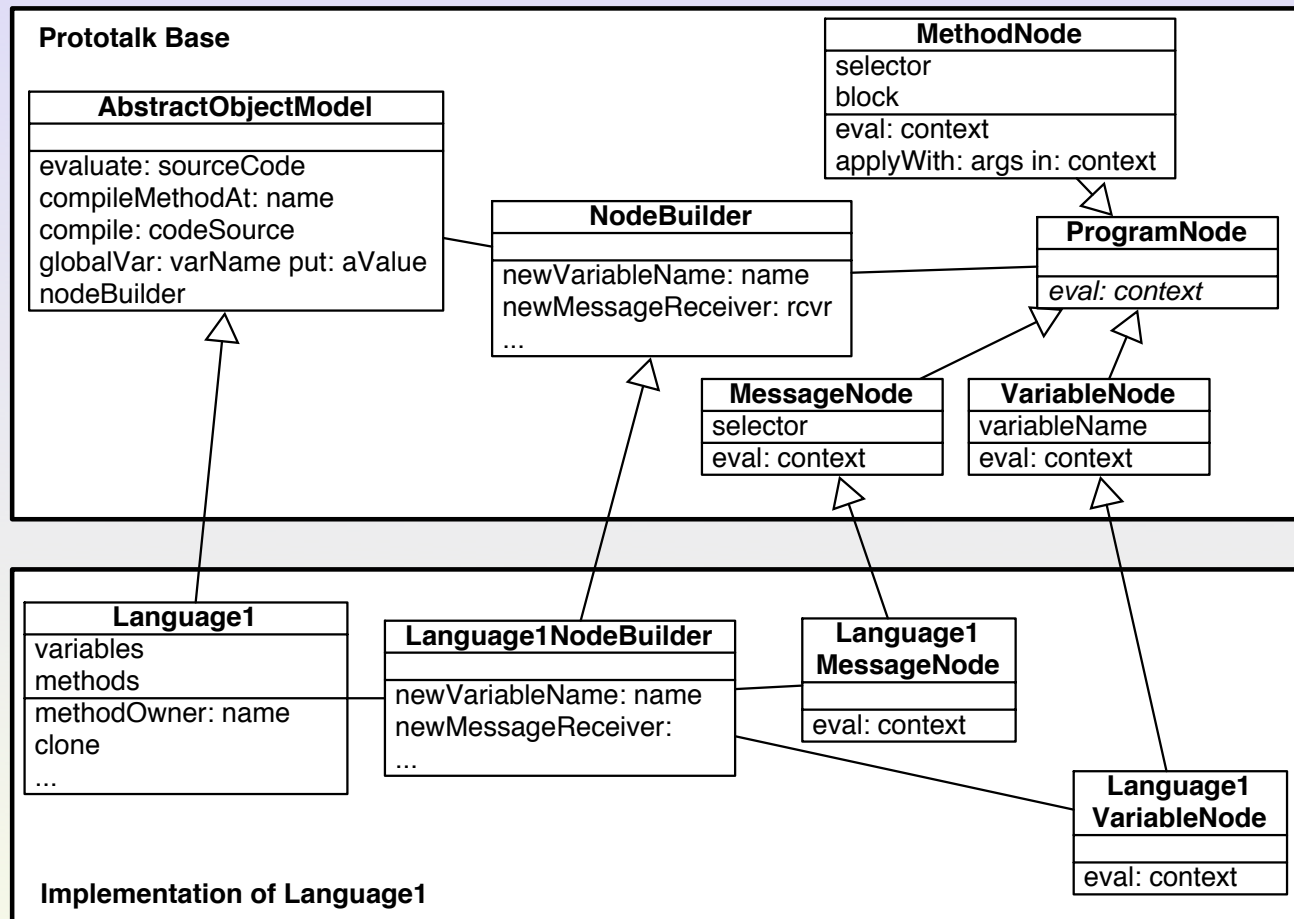
Goal and Design Decisions

- **Uniformity:** same syntax for all the languages
- **Minimality:** minimal kernel to express the most general semantics
- **Extensibility:** new languages has to be implemented efficiently and easily
- **Usability:** integrated in the environment

Prototalk

- Language represented by a subclass of **AbstractProto**
- Execution of program by sending **evaluate:** to it
- The AST is built using a **ProgramNodeBuilder**
- It is composed of subclass of the Smalltalk AST nodes

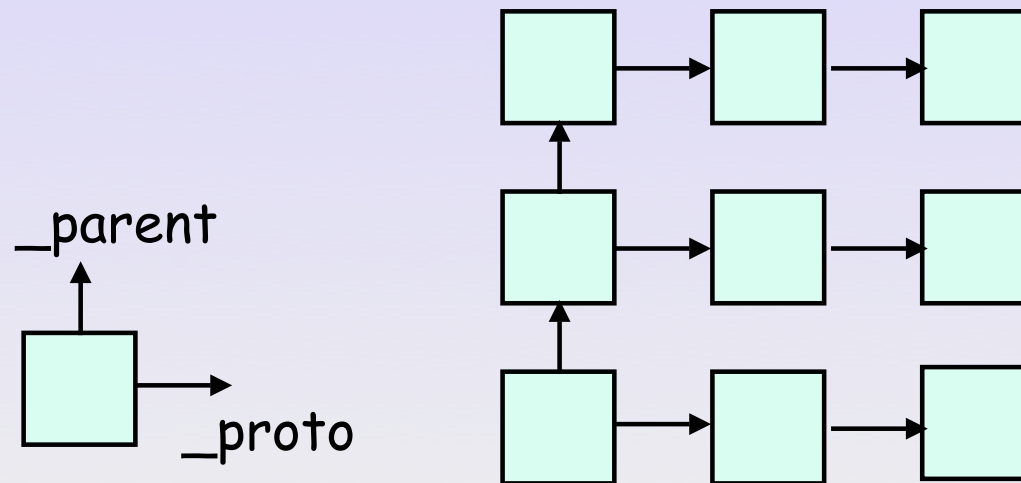
Architecture



Language1

- Language1 is a subclass of AbstractProto
- Ability to execute programs
- Prototypes are represented by its instances
- Provides primitives like clone, addVar: addMethod:, hasVar:, ...

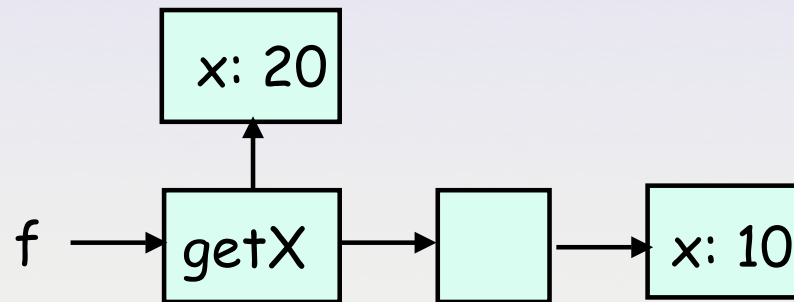
Example: NewtonScript



- It has a double inheritance: `_proto` is searched prior to `_parent`

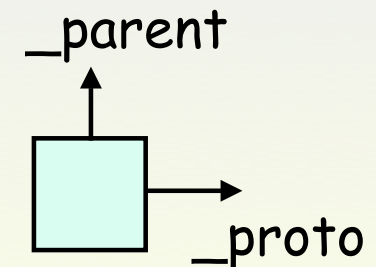
Double Inheritance

```
f := {getX: funct() x, _proto: {_proto: {x: 10}}};  
f._parent := {x: 20};
```



`f.getX() ==> 10`

||



In Prototalk

```
f := {getX: funct() x, _proto: {_proto: {x: 10}}};  
f._parent := {x: 20};  
f.getX() ==> 10
```

The same program in Prototalk:

NewtonScriptLike evaluate: '

 f := PRoot clone.

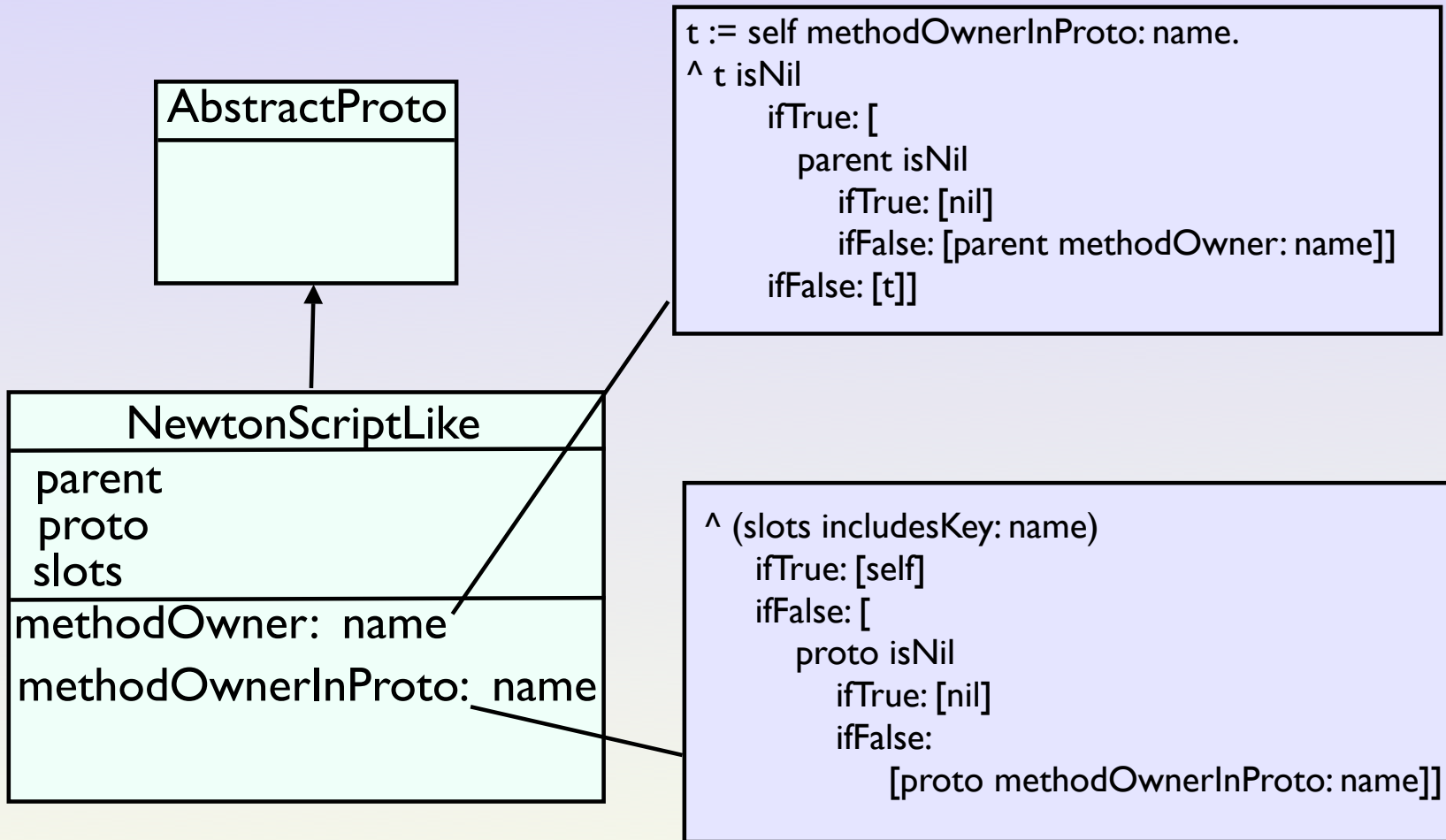
 f proto: (PRoot clone; proto: (PRoot clone; addSlot: "x = 10")).

 f addSlot: "getX x".

 f parent: (PRoot clone; addSlot: 'x = 20').

 f getX.'

In Prototalk

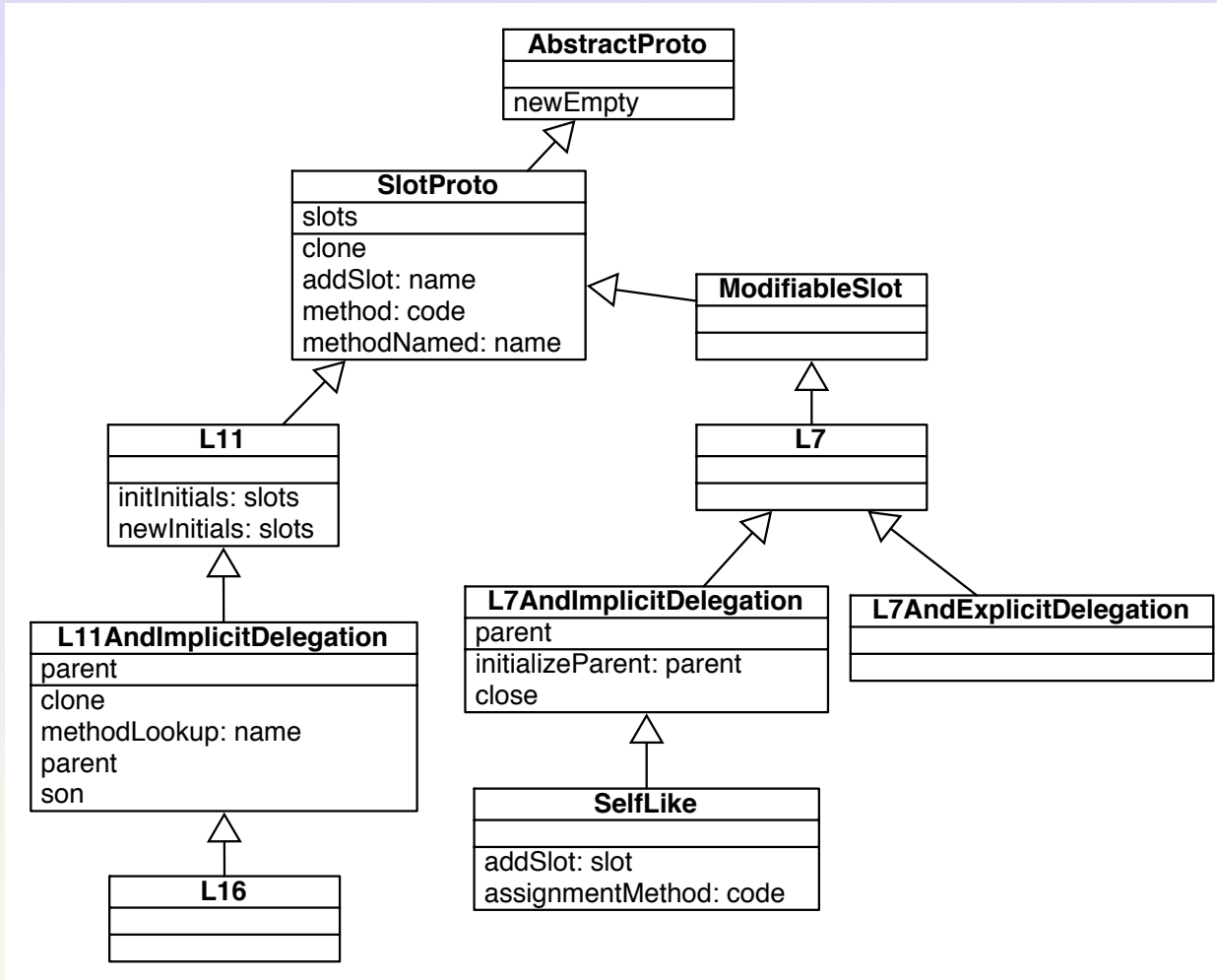


Conclusion

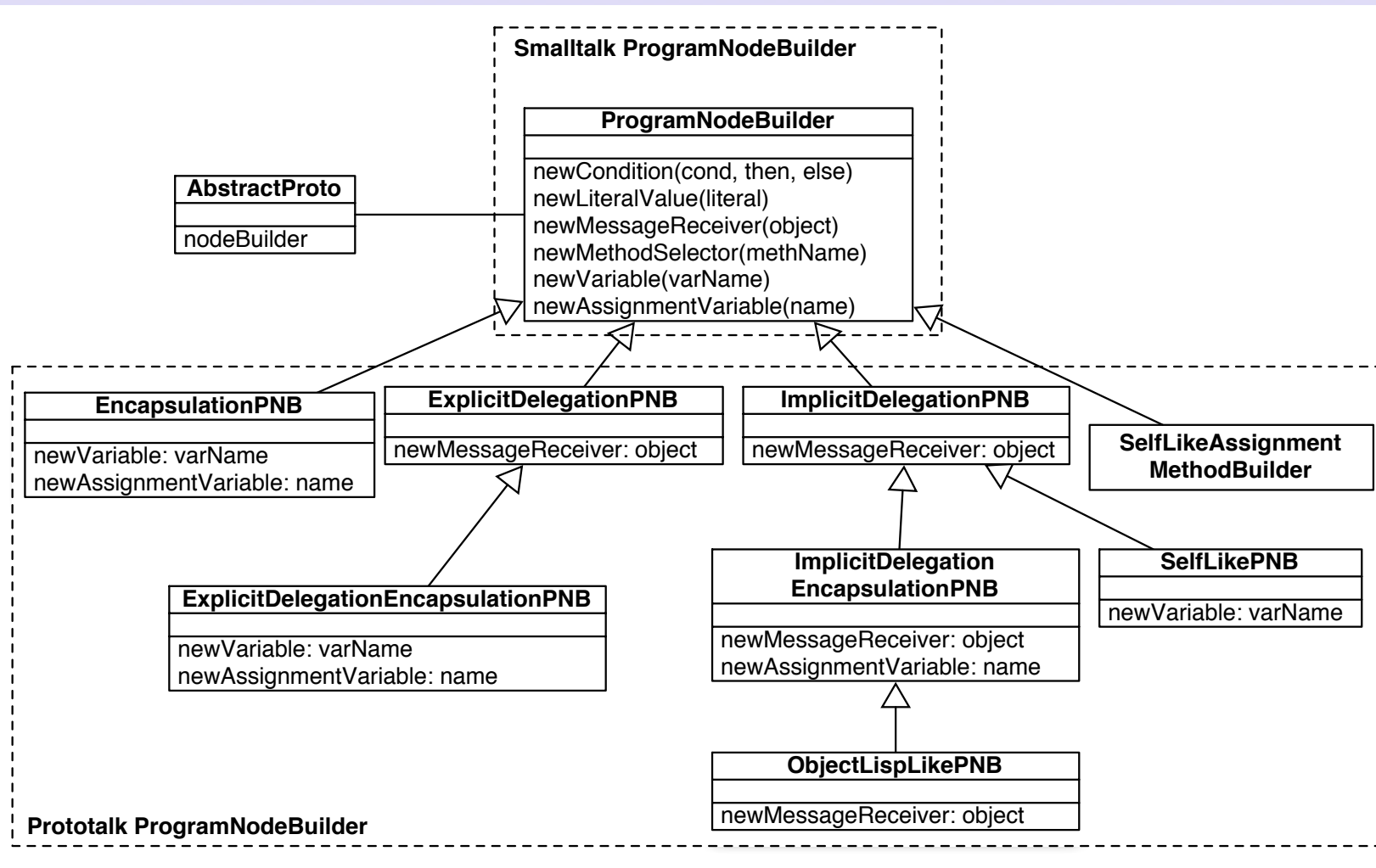
- Platform to design and experiment prototypes-based OO language
- Pure interpretation of programs
- Pedagogical tool: used at University Paris VI, Montpellier and Berne

- Prototalk: an Environment for Teaching, Understanding, Designing and Prototyping OO languages
- bergel@iam.unibe.ch

Hierarchy of Object Model



ProgramNodeBuilder



Self, ObjectLisp, ...

