

# Formal Integration of Generation Tools

Anthony M. Sloane  
Macquarie University  
Sydney, Australia

## Generative Programming using Off-the-Shelf Tools

- The idea is to save on overall development effort by reusing existing generation tools.
- Add value by "black box" integration.
- E.g., Eli (Colorado, Paderborn, Macquarie) in the domain of language processor generation:
  - 15MB installed, comprising 15+ tools.
  - Lexing, parsing, tree walking, symbol tables, unparsing, structured output.

STS Workshop 2004

Anthony Sloane, Macquarie University

2

## The Integration Problem

- The "black box" generation process integrates the individual tools:
  1. User specifications are processed to obtain tool inputs.
  2. Tools are invoked to produce code fragments or more specifications.
  3. Generated code is combined to form components of a program.
- How do we describe this kind of integration?

STS Workshop 2004

Anthony Sloane, Macquarie University

3

## Tool Integration Approaches

- Old:
  - Automatic build system a'la Make.
  - Ad-hoc shell scripts or custom programs to massage specifications or tool outputs into tool inputs.
- New:
  - Software transformation techniques to specify integration schemes.

STS Workshop 2004

Anthony Sloane, Macquarie University

4

## Describing Tool Integration with Software Transformation

- Transformation rules formally describe how specifications, tool inputs, tool outputs and code fragments relate.
- Also describe tool invocation points.
- Advantages:
  - A single paradigm rather than spread across build system, scripts and custom programs.
  - Better documentation of generation process.
  - Formal semantics leads to more reliable outcome.

STS Workshop 2004

Anthony Sloane, Macquarie University

5