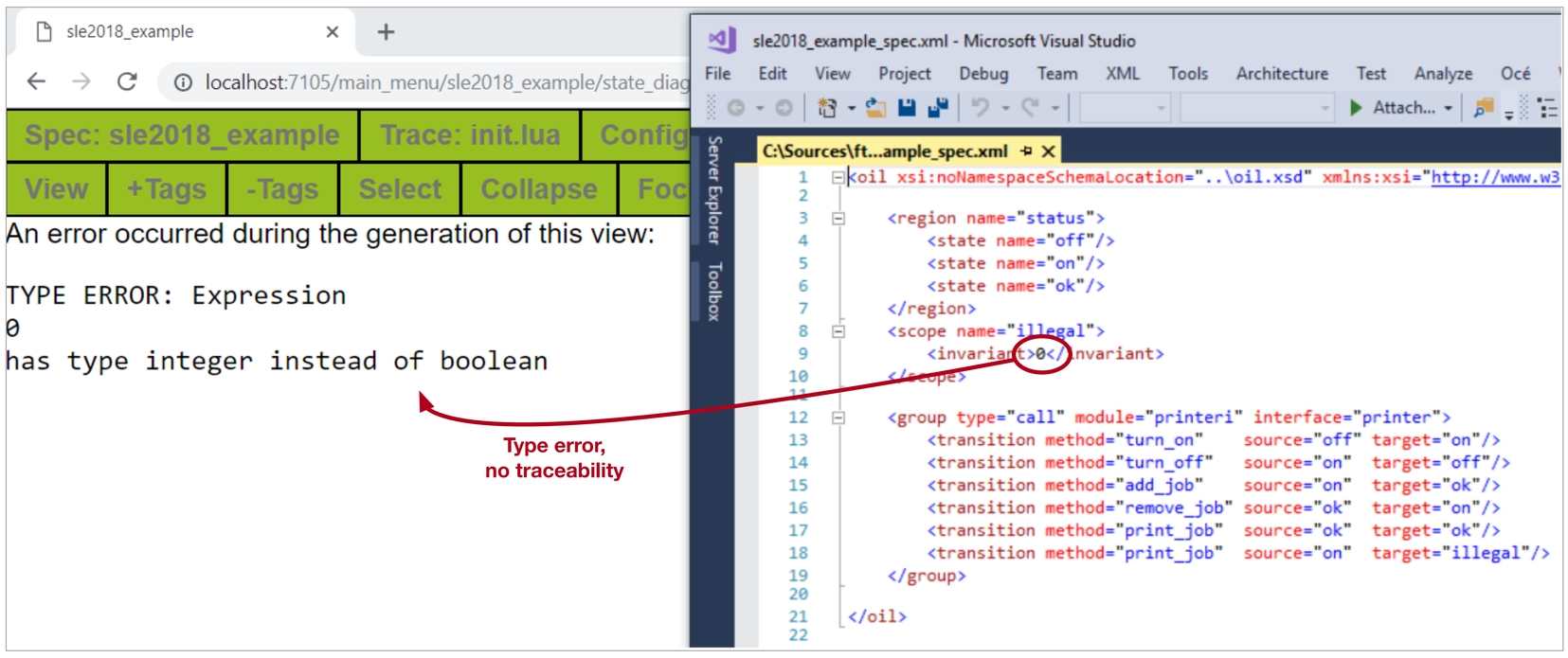
Migrating Custom DSL Implementations to a Language Workbench (Tool Demo)

Jasper Denkers
Delft University of Technology
Delft, The Netherlands
j.denkers@tudelft.nl

Louis van Gool
Océ Technologies B.V.
Venlo, The Netherlands
louis.vangool@oce.com

Eelco Visser
Delft University of Technology
Delft, The Netherlands
e.visser@tudelft.nl

Custom DSL Implementations



Context: Océ develops and uses domain-specific languages (DSLs) for model-based development.

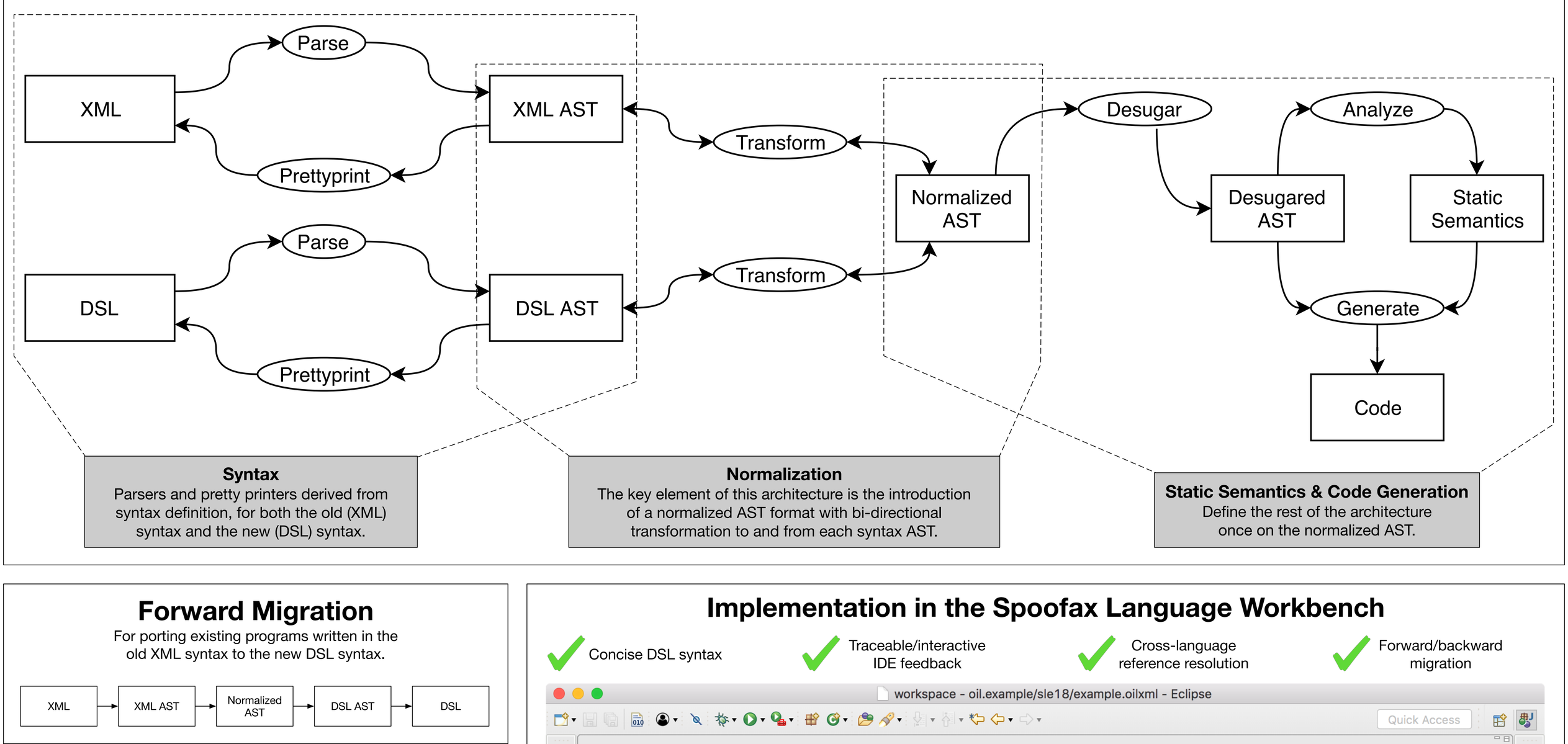
Problem: custom implementations using conventional technologies (XML for syntax, Python for static analysis and code generation) are flexible, but:

- ✗ No IDE support
- ✗ Hard to implement advanced language features
- ✗ Concise syntax is missing
- ✗ No traceable error reporting

Solution: migrate to a language workbench.

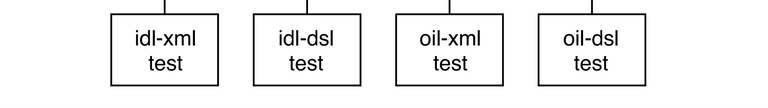
Extra requirement: automate forward and backward migration.

DSL Migration Architecture



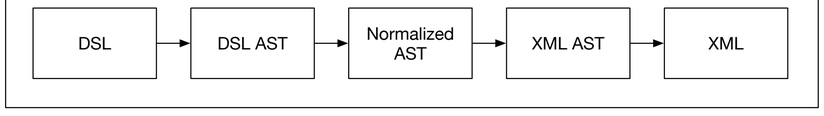
Forward Migration

For porting existing programs written in the old XML syntax to the new DSL syntax.



Backward Migration

For programs written in the new DSL to be backward compatible with existing tooling still based on the XML syntax.



Modular Language Definition

Inter-language: XML and DSL variants of the languages share syntax and static semantics definition.

Intra-language: OIL re-uses syntax, static semantics, and transformations from IDL.

Implementation in the Spoofax Language Workbench

✓ Concise DSL syntax
✓ Traceable/interactive IDE feedback
✓ Cross-language reference resolution
✓ Forward/backward migration