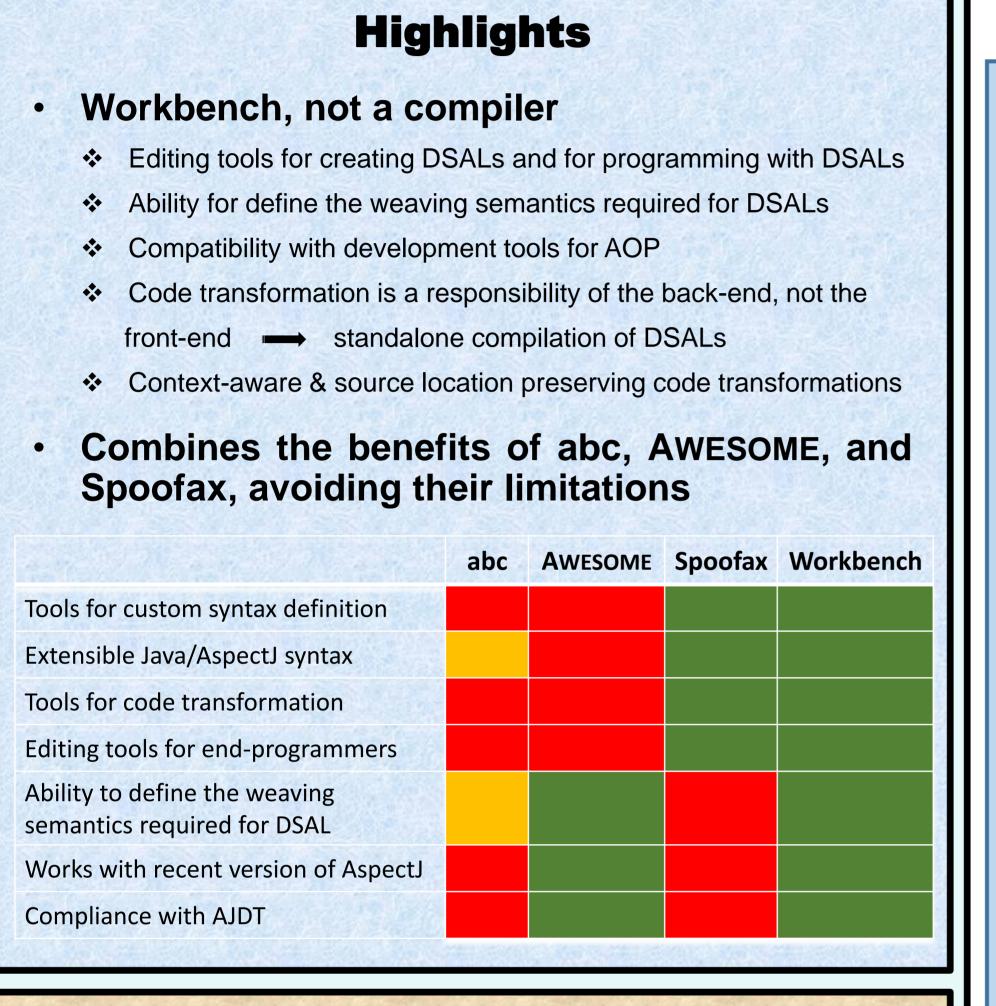
## A Language Workbench for Creating Production-ready Extensions to AspectJ

## Arik Hadas Open University of Israel

Research in domain-specific aspect languages suffers from the deterioration of the Aspect Bench Compiler (abc). We present an alternative language-oriented programmer's workbench for developing extensions to AspectJ. Much like the abc, the workbench allows researchers to implement and evaluate new extensions. In contrast to abc, however, it also provides IDE support, compatibility with AspectJ 1.7.4 and Java 7, and support for programming in multiple extensions. For validation we implemented the workbench by integrating Spoofax and AWESOME, and used it to create complex third-party extensions, including COOL, explicit join points (EJP), and closure join points (CJP).



## Validation

- Open source implementation
- Plugins for well-known third-part extensions
  - COOL, EJP, CJP

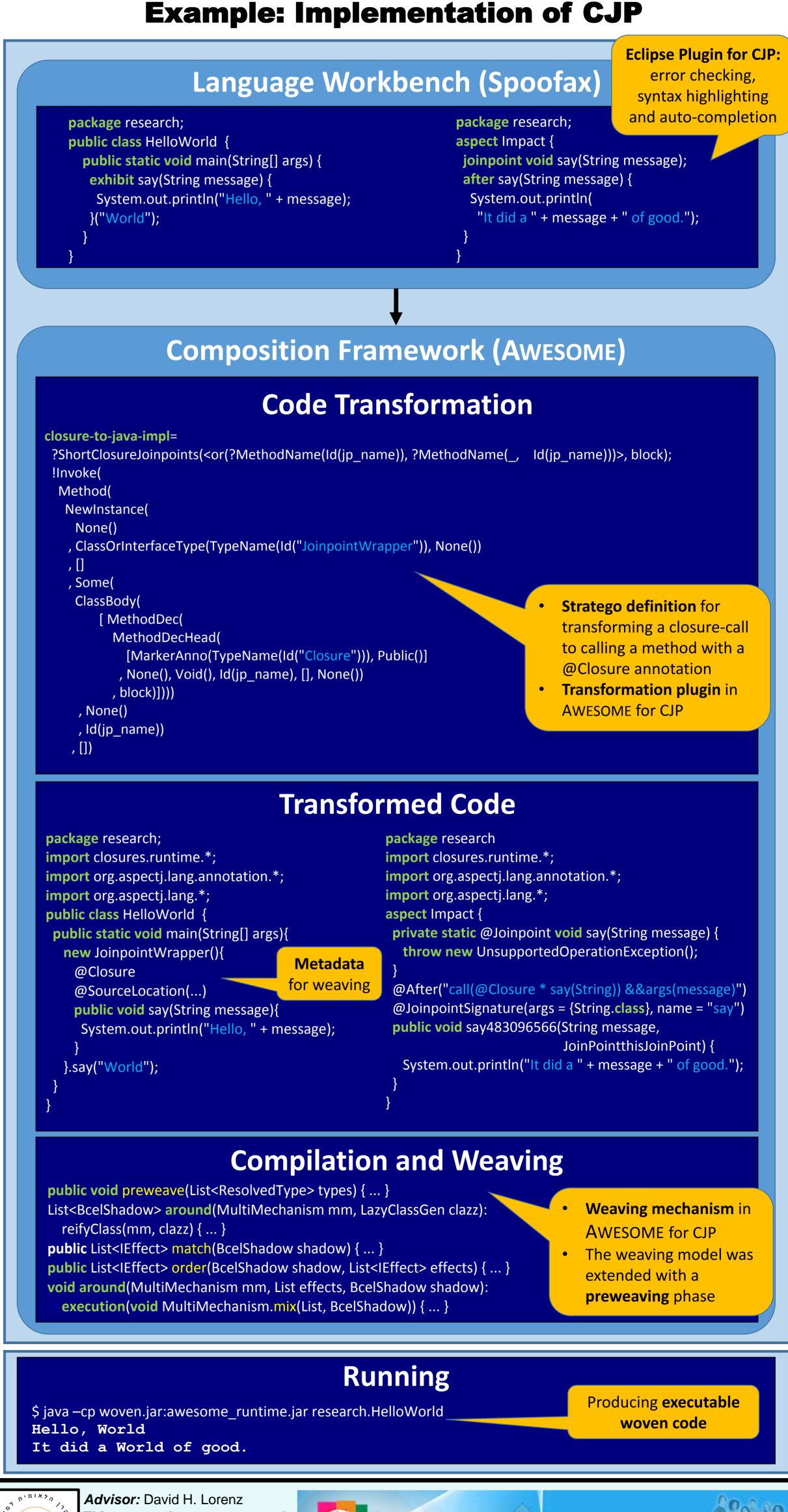
lexical syntax

"exhibit" -> Keyword

"joinpoint" -> PseudoKeyword

Including features that were omitted in original prototypes

## **Example: SDF for CJP** module languages/closures/Main imports languages/java-15/Main languages/aspectj/ajc/Main exports sorts JoinpointDeclaration context-free syntax "exhibit" MethodName "(" {FormalParam ","}\* ")" Block "(" {Expr ","}\* ")" -> Expr {cons("ClosureJoinpoints")} "exhibit" MethodName Block -> Expr {cons("ShortClosureJoinpoints")} JoinpointDeclaration -> AspectBodyDec "joinpoint" ResultType Id "(" {FormalParam ","}\* ")" Throws? ";" -> JoinpointDeclaration{cons("JoinpointDeclaration")} (Anno | MethodMod)\* CJPAdviceSpec Throws? Block -> AdviceDec {cons("CJPAdvice")} "before" Id "(" {FormalParam ","}\* ")" -> CJPAdviceSpec {cons("CJPBefore")} "after" Id "(" {FormalParam ","}\* ")" -> CJPAdviceSpec {cons("CJPAfter")} "after" Id "(" {FormalParam ","}\* ")" "returning" CJPSingleParam? -> CJPAdviceSpec {cons("CJPAfterReturning")} "after" Id "(" {FormalParam ","}\* ")" "throwing" CJPSingleParam? -> CJPAdviceSpec {cons("CJPAfterThrowing")} "(" FormalParam? ")" -> CJPSingleParam {cons("CJPSingleParam")} ResultType "around" Id "(" {FormalParam ","}\* ")" -> CJPAdviceSpec {cons("CJPAround")}





Advisor: David H. Lorenz
This research was supported in part by the Israel Science Foundation (ISF) under grant No. 1440/14.

