Evolution of Legacy Systems

Herman Tromp, Kris De Schutter
Department of Information Technology - University Ghent

Benevol2004 Workshop
Antwerp, July 2004

What is legacy ?

- Legacy :
  - Money, etc. (to be) received by a person by the will of and at the death of another person
  - Something handed down from ancestors or predecessors
- (source : dictionary)
-
- Legacy has a
  - Positive connotation : value
  - Negative connotation : old, out-of-date, inappropriate for current use

What is a legacy system ?

- A legacy system is an operational system that has been designed, implemented and installed in a radically different environment than imposed by the current ICT strategy. It is also a system that still contains considerable business value and is critical for operations.
-
- (Note : other definitions are available)

Topics in legacy evolution

- Programming languages (COBOL, C, OO languages, ...)
- Data storage (databases, flat files ,...)
- Inter process communication, CORBA, application servers, distributed systems, ...
- System architecture
- Analysis and design, UML, MDA, ...
- Re-engineering
- ...
- What will we be discussing specifically ?

Adding aspects to legacy code

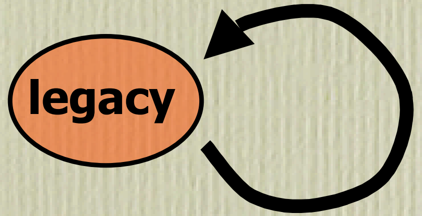
"Opposites are not contradictory but complementary."

-- N. Bohr

Adding aspects to legacy code

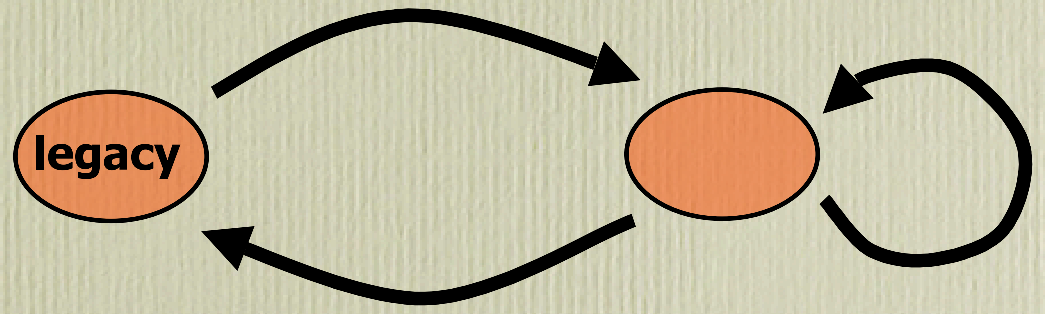
- There is no AspectCobol, AspectFortran,...
- There is an AspectC, but it is not (generally) usable
- So we have to set these up ourselves
- And do it as flexible as possible
  - we don't want to start from scratch every time we work on a new language (or dialect)
  - keep things open-ended as much as possible

A direct approach



Adding aspects by transforming the legacy sources directly

An indirect approach

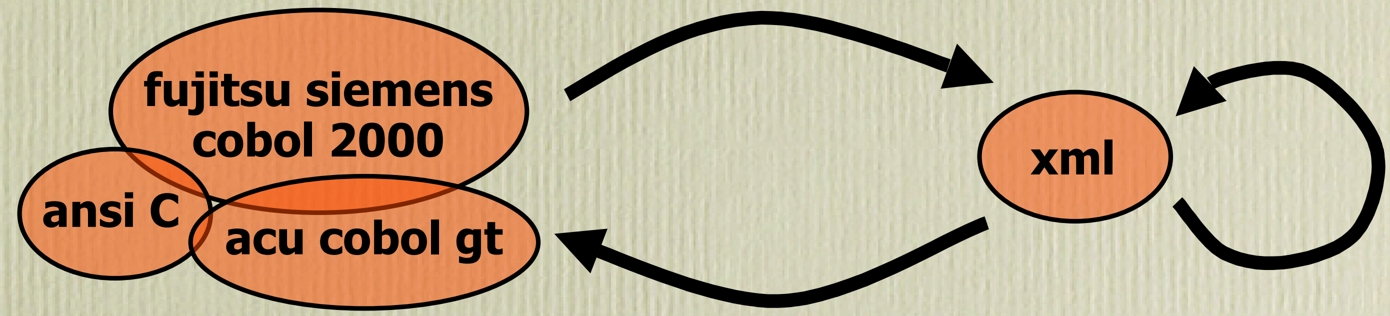


Transform the code into an intermediate format

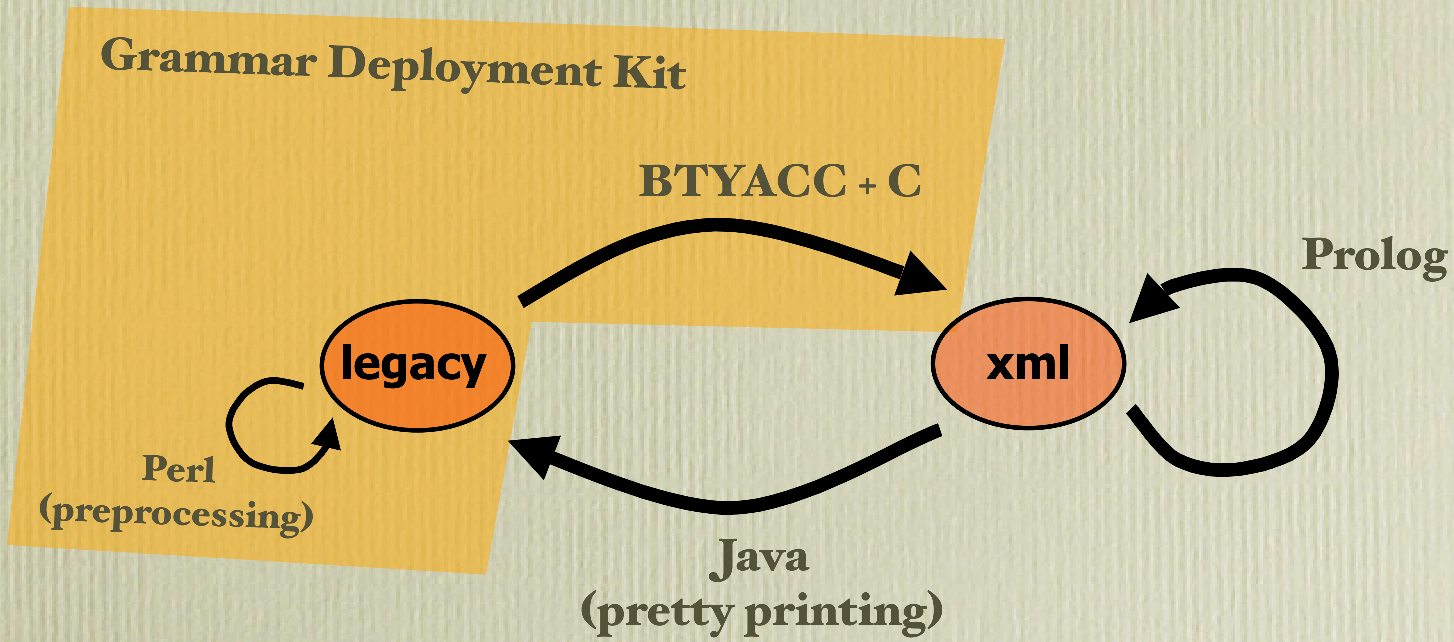
Add aspects by transforming the intermediate version

Transform back to the original dialect

In practice



Tools used


Grammar Deployment Kit
BTYACC + C
Prolog
legacy
xml
Perl
(preprocessing)
Java
(pretty printing)