How to make a bridge between transformation and analysis technologies?

TrafoDag Session

April, 2005
There is a **difference** between analysis and transformation

**Technological diversity:**
- Transformation: graph/term rewriting, fp, templates, . . .
- Analysis: databases, logic, relations, state machines, . . .

**Exploiting the benefits of understanding a domain**

**How to make a bridge between these kinds of technologies?**
You need unique identification of entities

Choice: passing source code locations or entities
  - Identify by reference: locations
  - Identify by value: unique id

Obtaining unicity can be a non-trivial computation

Locations are highly unstable
There is no **conceptual** problem.

Interesting engineering issues, inventoried by example.

Integration scheduling

- Lazy — get data when needed, or transform when needed
  
  SOUL, Stratego

- Eager — serialize and transmit full model
  
  TXL/GROK, ASF+SDF/RScript, Columbus

Composition: monolithic vs Separated

Language design:

- one general formalism for both
- separation into different domains
- conclusion: all existing systems separate the domain
Domain summary

The given two soups of technologies

- Computation
  - View
  - Analysis
  - Transformation
  - Bridge

- Render
  - Definition
  - Recovery
  - Synthesis
  - Presentation
  - Abstraction
  - Normalization
  - Optimization
  - lazy / eager
  - Locality

- local
- global
The variability without the constraints

Domain summary

Communication

Protocol

Co-routines-RMI
Batch-Pipe/Filter

Direction

Who's the Driver

Granularity

Identification

Marshalling

Representation

Data

1 way/2 way
left/right

AS \{T, G\}

Architecture
Call graphs

By reference
By value

Get/Put
Linking
Sharing
Files

AS \{T, G\}

Parse trees
Separating versus Unifying

Monolithic approach

- Advantages
  - Consistency is easier
  - No marshalling

- Disadvantages
  - Tends to loose domain benefits
  - Glass boxes, everything depends on everything

Separated approach

- Advantages
  - Forces explicit contracts
  - Reuse, variability by composition

- Disadvantages
  - Marshalling may be complex
  - Update consistency problem
Identity

Identified features and variability

Discussed some trade-offs

My suggestion for the future:

- Fill out the variability analysis and formalize in FDL
- Then express constraints that capture trade-offs (implication, exclusion, dependance)