Kahn Process Networks
Reconfiguration and Distribution

Julien Vayssière
Darren Webb
Andrew Wendelborn
Outline

Motivation:
"process composition" & metacomputing

Kahn Process Networks: basic concepts

Reconfiguration: a “node-centric” view

Representation: a Java API

Implementation: ProActive

Distribution: PAGIS

Future work

Distributed (storage) deadlock detection

Meta-programming and adaptive reconfiguration
PN Model

- Directed graphs (possibly cyclic)
- Nodes represent transformations
- Edges represent flow of data between nodes
- Edges are unbounded
- Two communication primitives: get (blocking) and put (non-blocking)
- Implicit parallelism
- Determinate
Fibonacci PN

Examples

1

1

Conversion

Rectification

Conversion

add

cons

cons

add
Formal Semantics

Characterizes behaviour of system
Equations over histories
Processes define history functions
Channels carry histories
Solution is least fixed point of system
Unique, exists, and can be found iteratively
Node can be a recursive process schema
As we have semantics of reconfiguration
Unfolding of recursive schema
Alternative realizations of same history function
Reconfiguration

Reconfigured into PN node defn
\[ \text{el } c1, c2; \]
\[ \text{P1(out c1);} \]
\[ \text{el-centric} \]

This reconfiguration by *substitution* at *implementation* level
PNs in Practice

Processes (nodes) are “well-behaved” and can be programmed in any host language. In practical terms, there are many ways to define constituent computations, many ways to schedule these computations, and we can utilize a variety of computational engines. Different computational models include demand/data driven; hybrid examples: coroutines; Unix processes; PAGIS; DF
Using PNs

continue by looking at aspects of
Building a PN
• Coordination language specifies plumbing (connections)
• GUI; textual language; Java API

PN representation
• Consistency under reconfiguration

Executing a PN
• A specific computational engine (CE)
  – Processing and scheduling
Other Work

Kahn and MacQueen; Wendelborn; Böhm
- Declare, create and configure channels
- Less general view of reconfiguration

DISCWorld
- Coarse-grain “dataflow”
  - Placement of computation & deferred delivery of results

Lee et al (Ptolemy II, UCB)
- General entity/relation model
- Mutable workspace for representation consistency
  - Centralized reconfiguration / deadlock detection

Moses
- Model description and interpreter (PN model)