

DFL designer

Collection-oriented scientific workflows
with Petri nets and nested relational calculus

Jacek Sroka
Piotr Włodarczyk
Łukasz Krupa

*University of Warsaw,
Poland*

Jan Hidders

*Delft University of Technology,
The Netherlands*

Presentation plan

- Introduction
 - Collection-oriented scientific workflows
 - Petri net
 - nested relational calculus
- DataFlow Language (DFL)
 - Why yet another model, why yet another tool?
- DFL designer
 - Integration with Taverna
 - Presentation of features

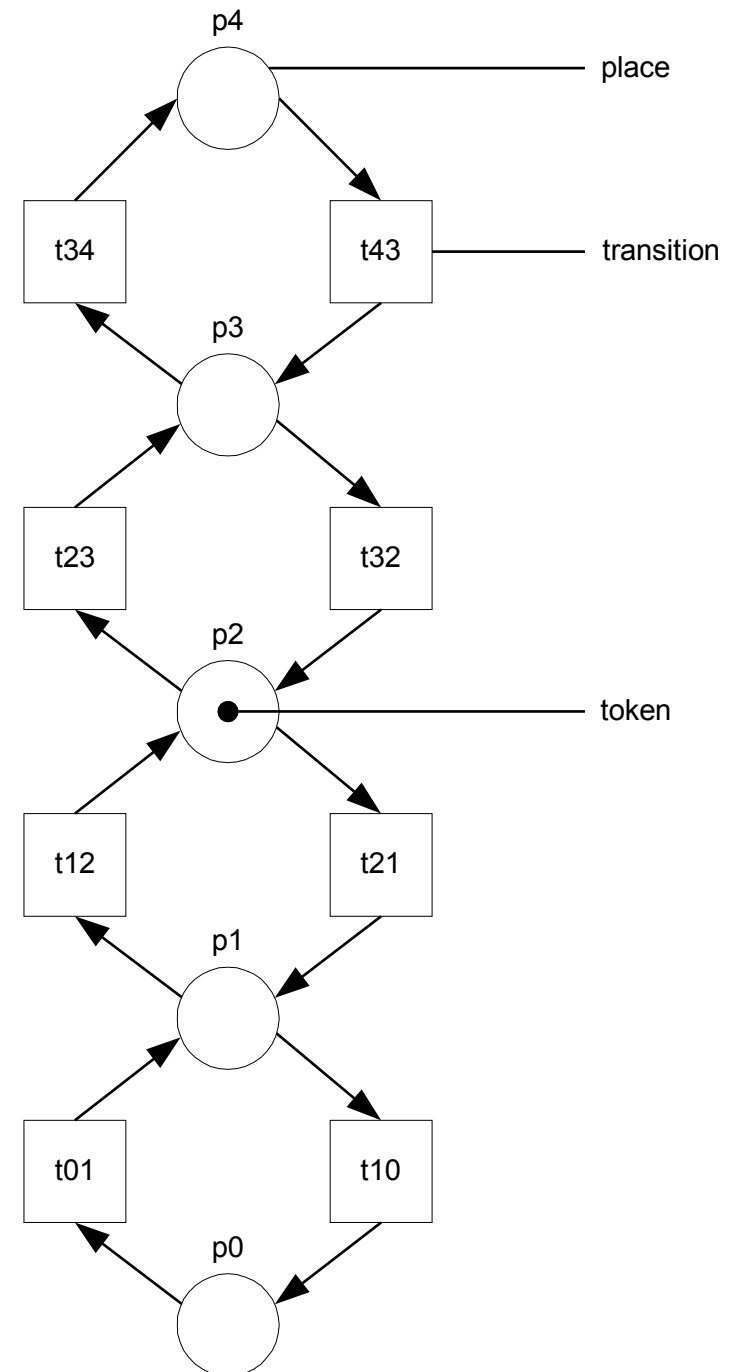
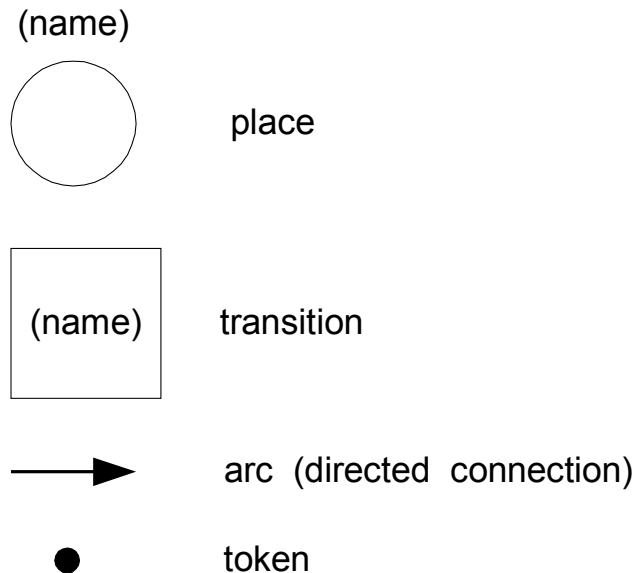
Collection-oriented scientific workflows

- **Workflow:**
(based on Wikipedia) Workflow software aims to provide end users with an easier way to orchestrate or describe complex processing of data in a visual form, much like flow charts but without the need to understand computers or programming.
 - distributed systems represented graphically as graph/net
- **Collection-oriented:** processing of large amounts of structured data
- **Scientific:** applied in life sciences
 - ecology, geology, chemistry, astronomy and especially **bioinformatics**
 - *in silico* experiments as opposed to *in vitro* (saves money and time)
 - life sciences introduce new challenges

Petri nets

Carl Adam Petri (1962, PhD thesis)

- Places and transitions
- Directed connections
- Only nodes of different kind can be connected
- Places may hold zero or more tokens



Practical presentation

Is this relevant

- Do we have side-effects?
- How do users think about this?
- What language do they really need?
- What results they want to reuse?

- Space flight

- Banking

- Tailoring

What about the data

- Petri nets cover only the control flow
- Nested Relational Calculus (NRC)
 - Theoretical model for XML and object-oriented query languages
 - Types: basic types, records, nested collections
 - Operations: typed data constructors/deconstructors, **map**
 - Extensible

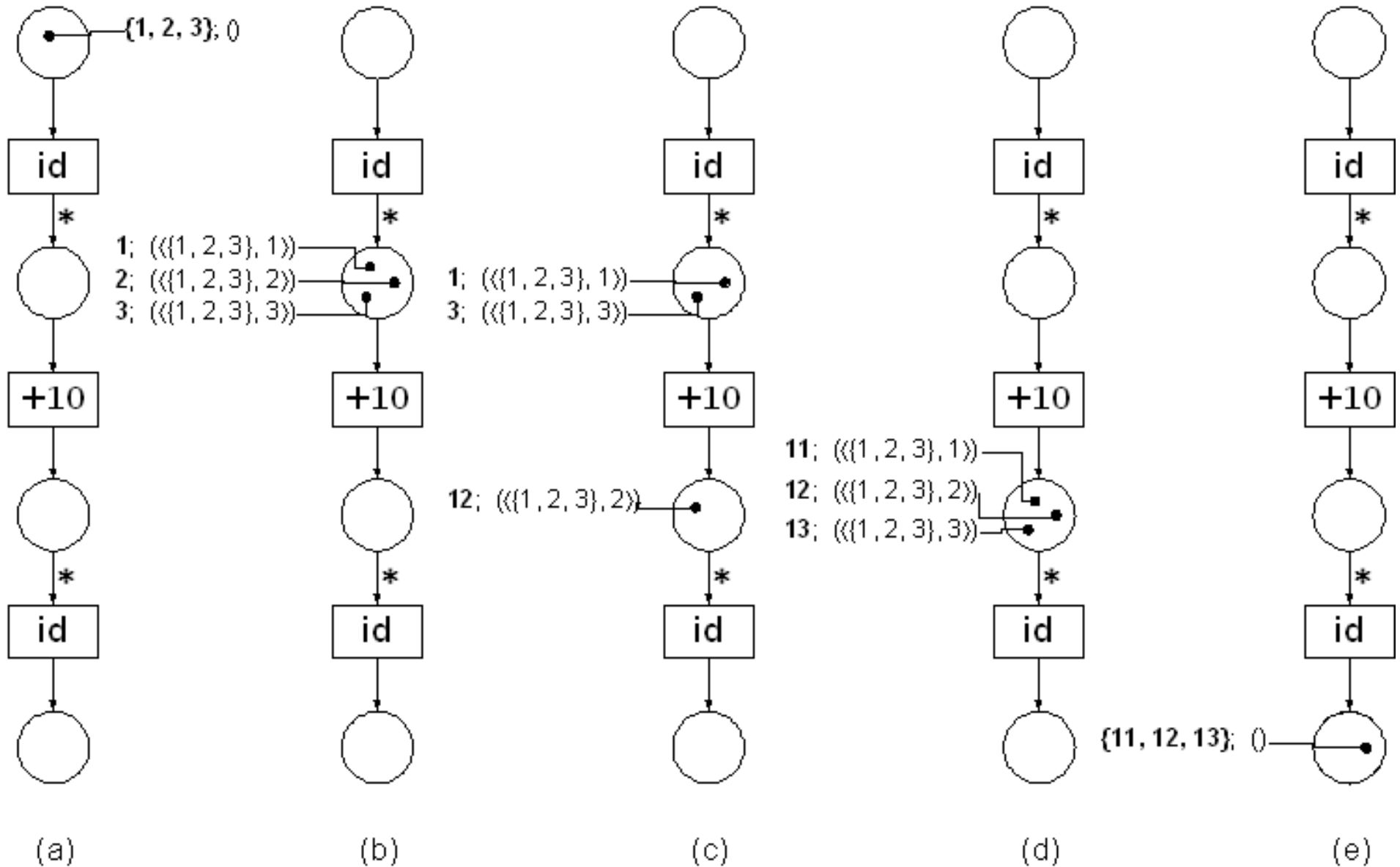
Advantages of NRC

- Optimization (NRC -> XQuery, Linq)
 - If no side-effects, then can be applied directly
- Distributed computation (NRC -> MapReduce)
- Data integration (Limsoon Wong, BioKleisli, HGP)
- Type inference
- Provenance models
- Assumes no side-effects

Petri nets + NRC = DFL

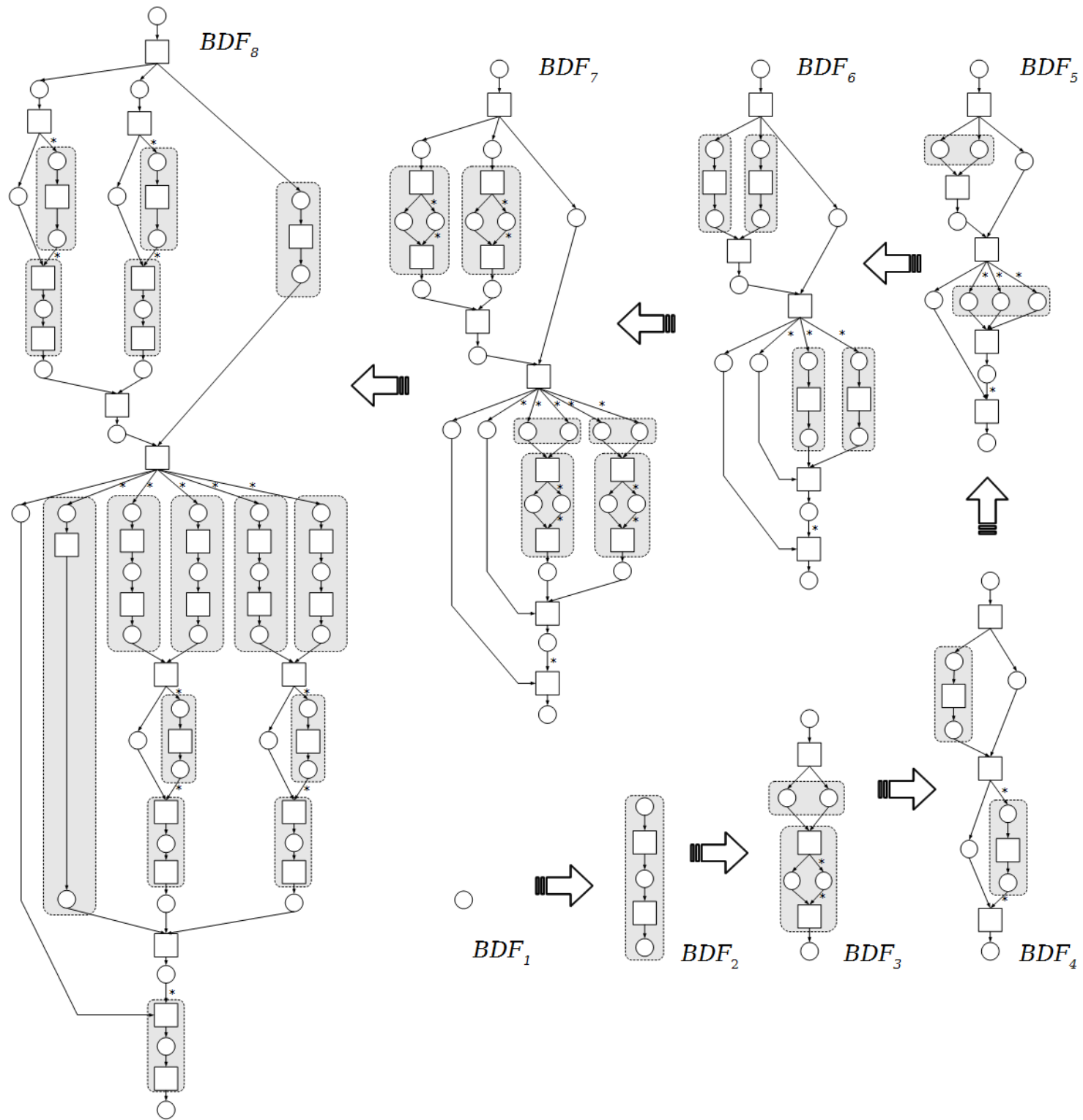
- Petri nets
 - control flow
 - graphical notation
- Nested Relational Calculus (NRC)
 - handling collections of data items (iteration)
 - typing system and operators
- DataFlowLanguage (DFL)
 - close to the principles
 - designed for easy analysis and verification
 - reuse of existing results is possible
 - can be used to define formal semantics for user-oriented COSW languages

Token unnesting history (kind of provenance)



Soundness and semi-soundness

- Classical soundness:
 1. One input token \rightarrow one output token
 2. One input token \rightarrow the computation can be always completed
 3. One input token \rightarrow all transitions can be fired
- (3) may depend not only on the structure of the net but also on the values and operations



Theorem

Hierarchical COSWs are semi-sound

- It is easy to check if COSW is hierarchical
- It is easy to transform into NRC (assuming no side-effects)

Practical presentation

Summary

- Combination of workflow formalism with a database formalism
 - Dataflow vs side-effects
- Creation of a tool with novel features
 - Token game
 - Easy debugging
 - Translation to NRC
 - Checking of correctness
- Linking with existing workflow engine
 - Reuse of services and visualizations
 - Real life examples
- TODO
 - Integration with NRC/XQuery engine
 - Integration with MapReduce framework

Questions?