Petri Nets for Reverse Engineering

Walter Keller

PhD thesis at Ifi, University of Zürich

Referee: Prof. Dr. Lutz H. Richter

Coreferee: Prof. Dr. Helmut Schauer

Clash

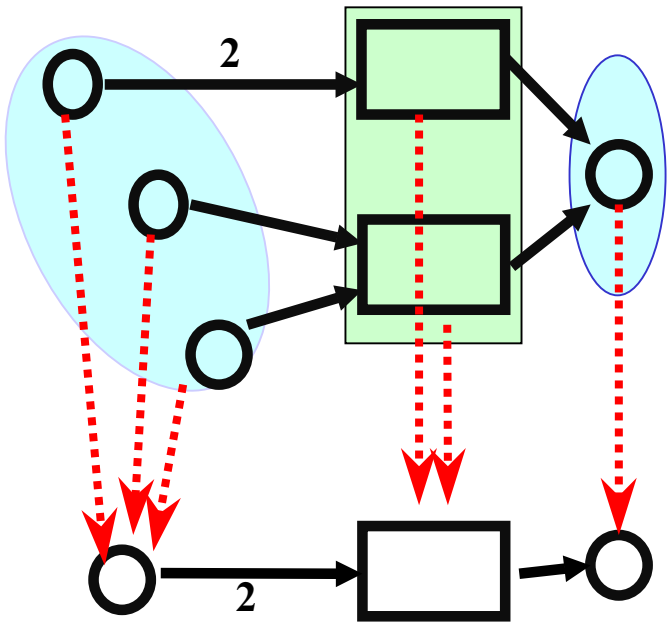
Petri nets

- strong in concurrency
- precise semantics
- lack of compositionality

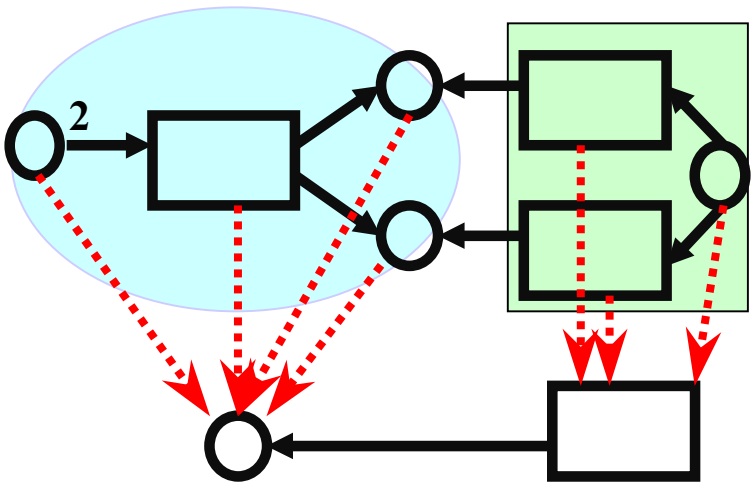
Reverse Engineering

- for conventional applications
- informal diagrams
- component analysis

Folding

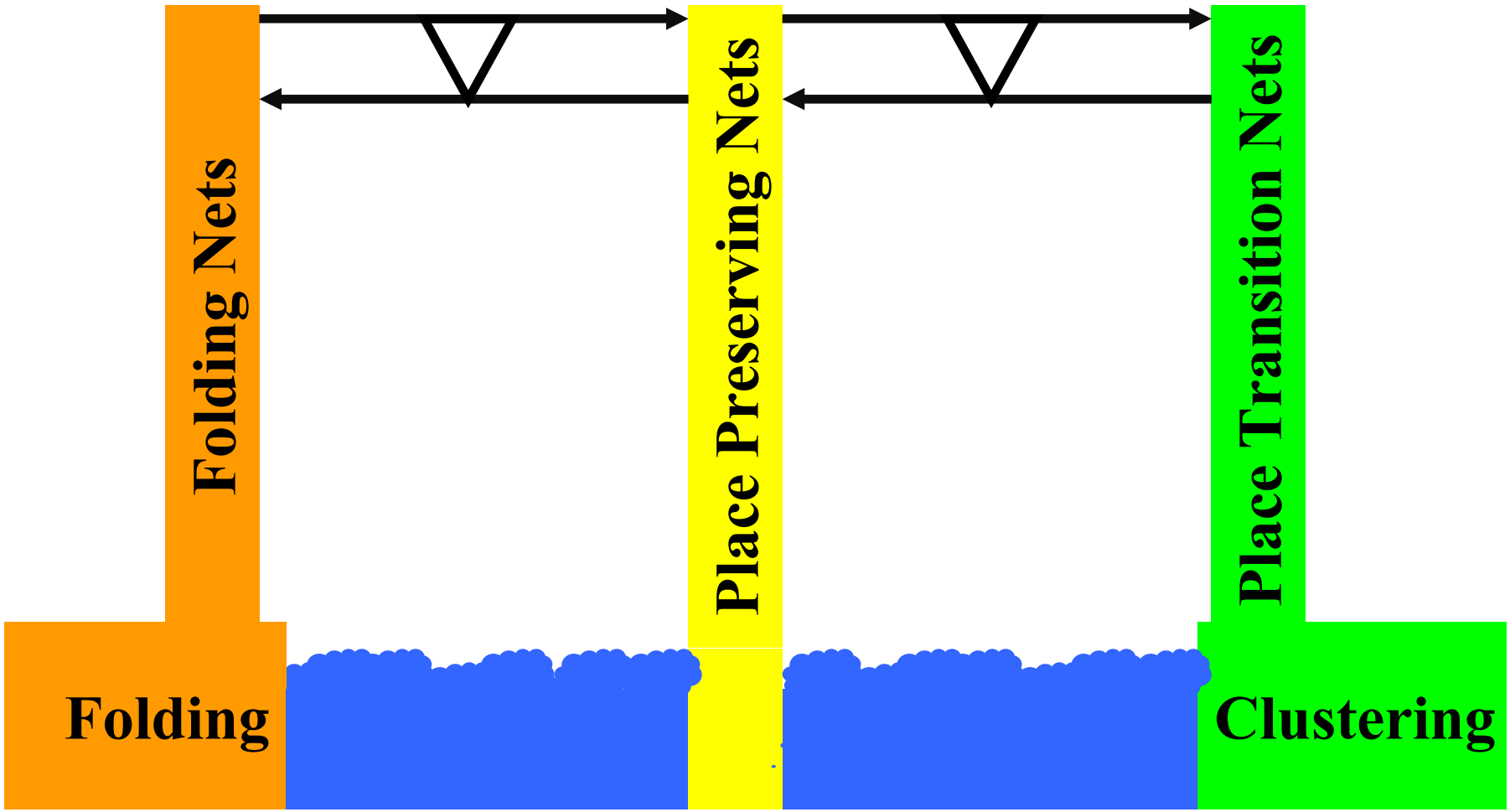


Clustering

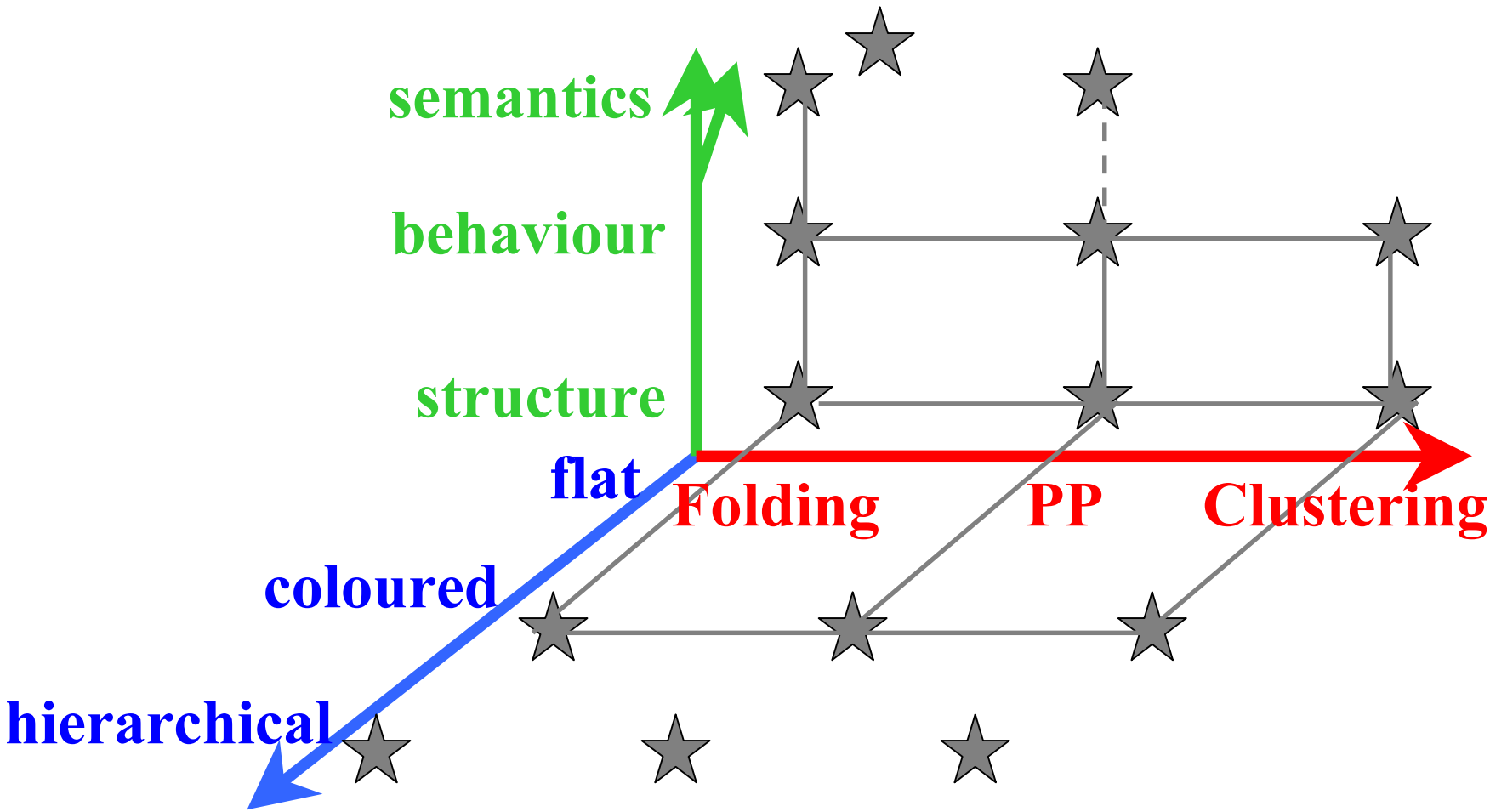


	Folding	Clustering
merges	• similar nodes • similar relationships	• neighboured nodes and relationships
preserves	• semantics	• locality
used for	• simulation	• system composition
typical for	• Petri nets	• engineering

The Bridge


Folding Nets
Place Preserving Nets
Place Transition Nets
Folding
Clustering

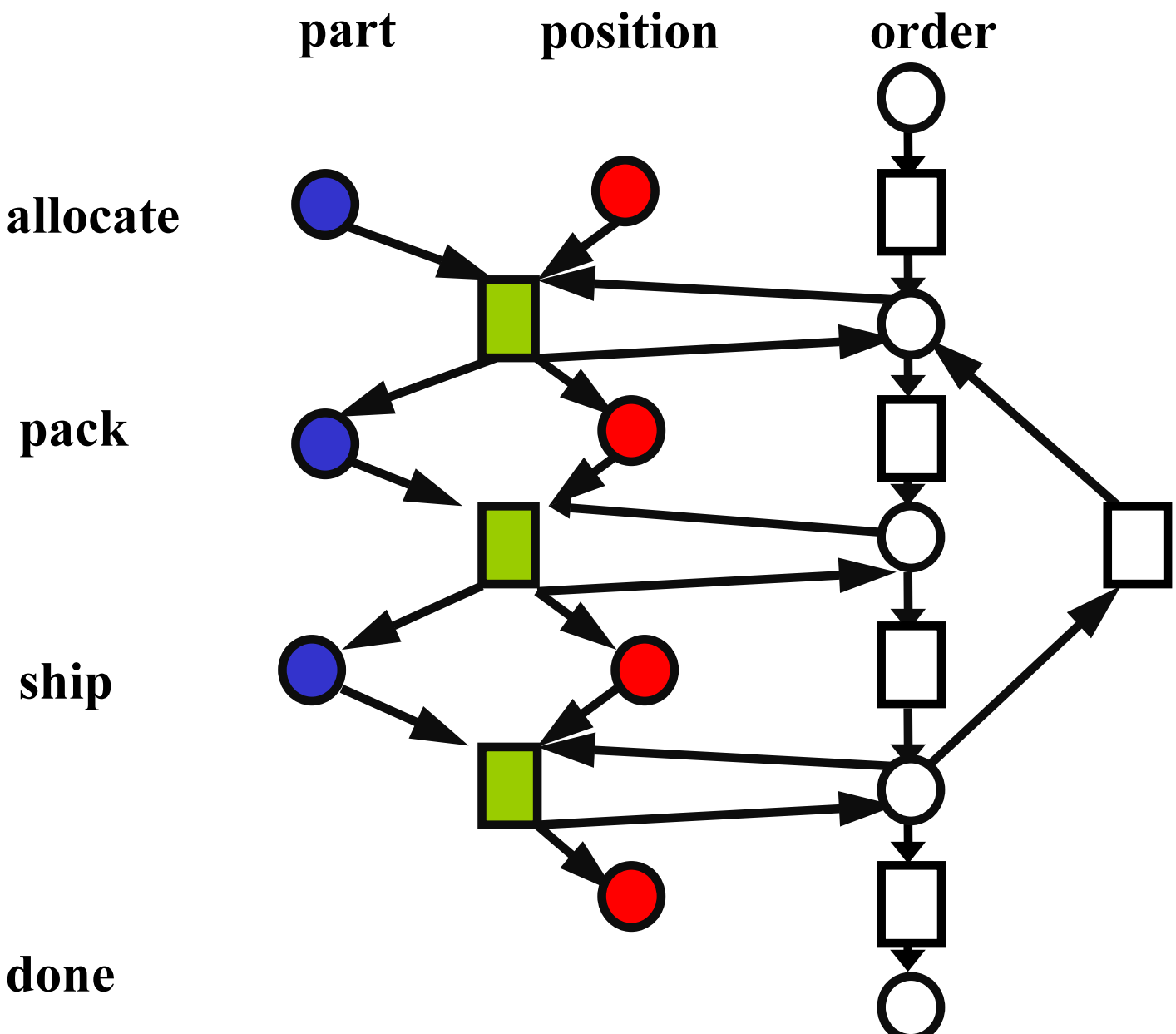
3 Axes of Variation



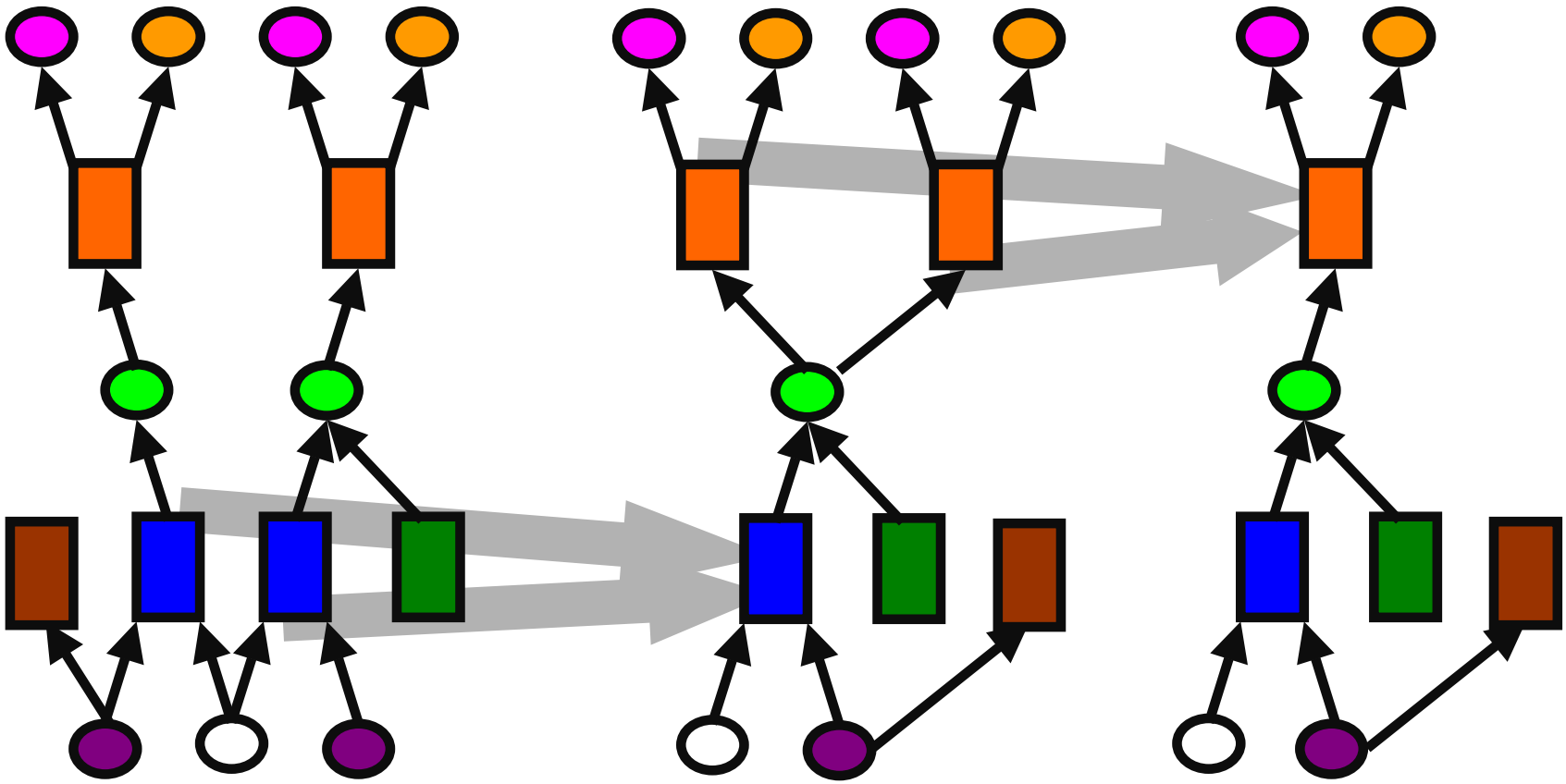
Petri-Net Models

- software engineering by folding
- clustering of structured programming terms
- data flow
- control flow
- dynamic traces
- Petri nets

Spare-Part System



Reductions



Reductions

- iterated coequalisers
  - of a set of morphism pairs
- maximal reduction has universal properties
- compute on single transition nets only
- variations in
  - similarity
  - neighbourhood
  - choice

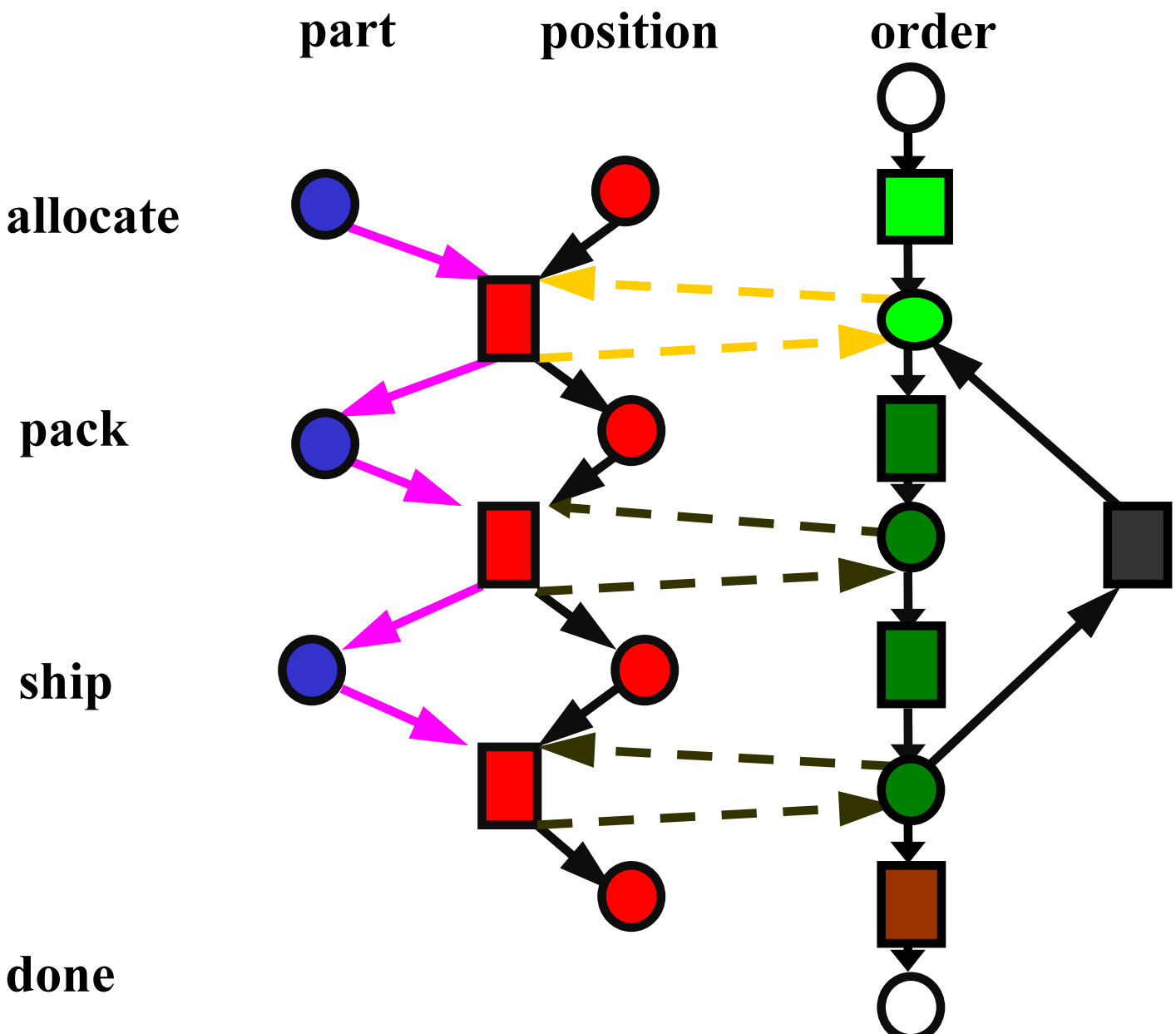
Reduction Algorithm

- unique maximal reduction
- often gives a compact and useful analysis
- good starting point
- allows many variations
- prototype in Smalltalk

Refine and Colour

- analyse relationships of a first reduction
  - split 1 and n sides of 1:n relationships
- second reduction
  - compatible with relationship analysis
- merge colour sets
  - along bijective relationships
- classify colour relationships
  - equality, composition, sub-relationship

Reverse-Engineered



Results

- Petri nets as an engineering metaphor
  - behaviour for intuition
  - bipartite graphs for real work
- folding- and net-based reverse engineering
  - new, fast and flexible algorithm
- Petri nets
  - categorical bridge from folding to clustering

Abstract

The goal of this work was to explore synergies between Petri net theory and reverse engineering. The result is a bridge from

- **clustering techniques - merging neighboured nodes which is a key feature for software engineering and the practical applications of Petri nets- to**
- **folding techniques - merging only transitions with transitions and places with places, preserving behaviour and allowing theoretical connections to many models of concurrency.**

To Petri net theory we contribute a new treatment of clustering. We introduce a category of Petri nets with morphisms that support clustering, offering attractive properties to software engineering and integrating smoothly with invariants. A computational reasonable adjunction connects it to folding based Petri nets – to two new cocomplete and complete categories. The dichotomy of structure and behaviour of Petri nets is expressed as compatible adjunctions to behavioural categories. Finally reachability and process semantics are attached categorically and a new variant of occurrence nets is proposed as a purer image of causality and branching.

For reverse engineering we model structural and functional aspects of software through Petri nets. Universal constructions within the above categories are able to recover high level design information from a flat net representing a low level implementation. This gives an algorithm with nearly linear cost. A prototype shows its value for reverse engineering and a rich palette of variations that allows to adapt to different situations. Finally we propagate Petri nets as a design metaphor for conventional software engineering.