A Framework for Interpreted Graph Models

Ivan Poliakov, Victor Khomenko, Alex Yakovlev
Interpreted Graph Models

• Static graph structure
  – Nodes
  – Arcs

• Additional entities
  – Tokens
  – Node states
  – Arc states
  – Etc

• Examples:
  – Petri Nets
  – Static Data Flow Structures
  – Gate-level circuits
Workcraft — objectives

• For researchers
  – define new Interpreted Graph Models
    (such as extended Petri Nets with inhibitor arcs, timing etc.)
  – inherit visual editing and simulation features from the framework

• For system designers
  – a consistent multi-formalism framework
  – convenient analysis and verification features
Petri Nets as low-level 'language'

- It is often not reasonable to develop special theory for higher-level models
- Petri Nets have richly developed theory and efficient tools
  - Petri Net Mapping approach: express high-level models in terms of PNs for analysis/verification
Technical details

- Framework is written in Java
  - Works cross-platform
- Plug-in based architecture
  - Plug-ins are Java Classes
  - Model types, node types, tools, import/export functions are plug-ins
- For computationally intensive functions (e.g. verification) Java tools are just an interface to external command-line tools
  - Highly efficient but still very flexible cross-platform operation
(live demo)
Use case — Static Data Flow Verification

Original state

29 steps

Deadlock state
Use case — Asynchronous circuit verification

Hazard caused by wire delay

wire delay

wire delay
Model interoperability/co-verification

Signal Transition Graph  
(Environment specification)

Gate-level circuit

Composition

Petri Net

Formal verification  
(PETRIFY, PUNF, MPSAT)
More complex interoperability
Conclusions

• Workcraft is a framework for Interpreted Graph Models
  – Provides visual editing and simulation features
  – Provides automated verification features

• Workcraft has been used for several practical applications
  – SDFS verification
  – Circuit verification (e.g. the design of a multiresource arbiter by Golubcovs et al.)
  – CPOG-based synthesis

Available free of charge for academic use at workcraft.org
End

Thank you!

Questions?