Simulation, Verification, & Automated Composition of Web Services

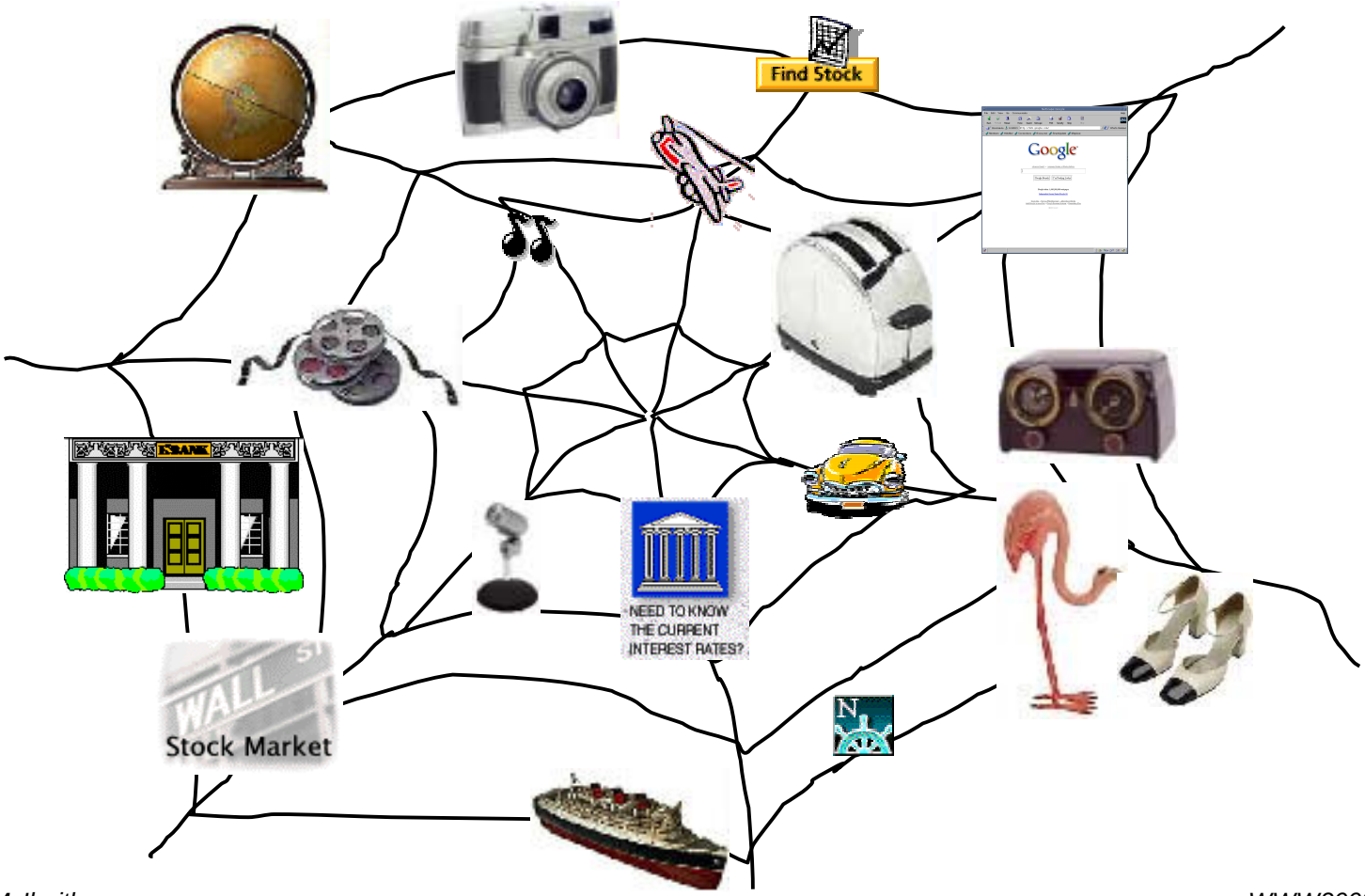
Srini Narayanan
ICSI & UC Berkeley
(formerly SRI)
snarayan@icsi.berkeley.edu

Sheila McIlraith
KSL, Dept. Comp. Science
Stanford University
sam@ksl.stanford.edu

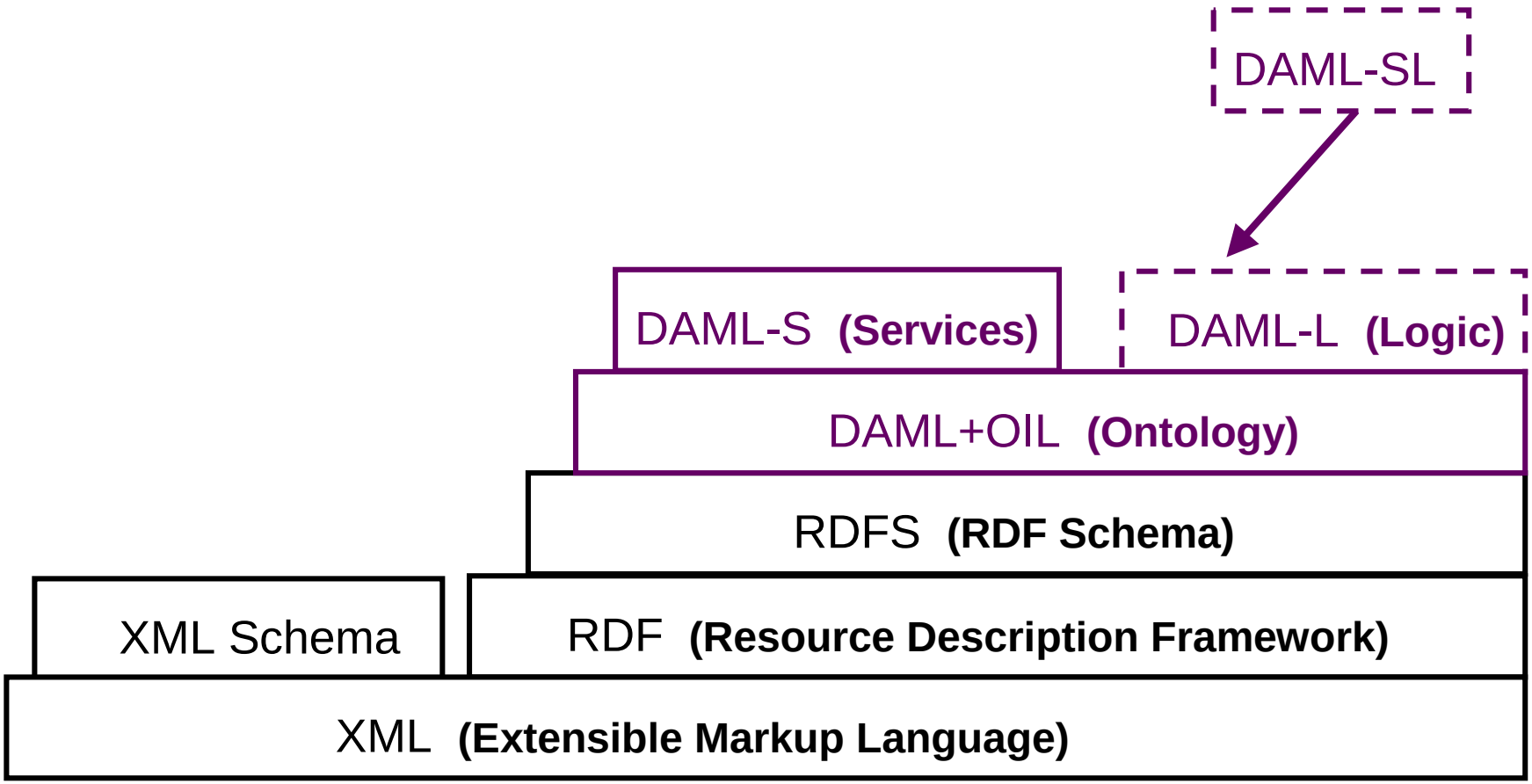
WWW 2002
May 8, 2002

Web Services

Web-accessible programs and devices



Knowledge Rep'n for "Semantic Web Services"



DAML-S: Semantic Markup for Web Services

DAML-S: A DARPA Agent Markup Language for Services

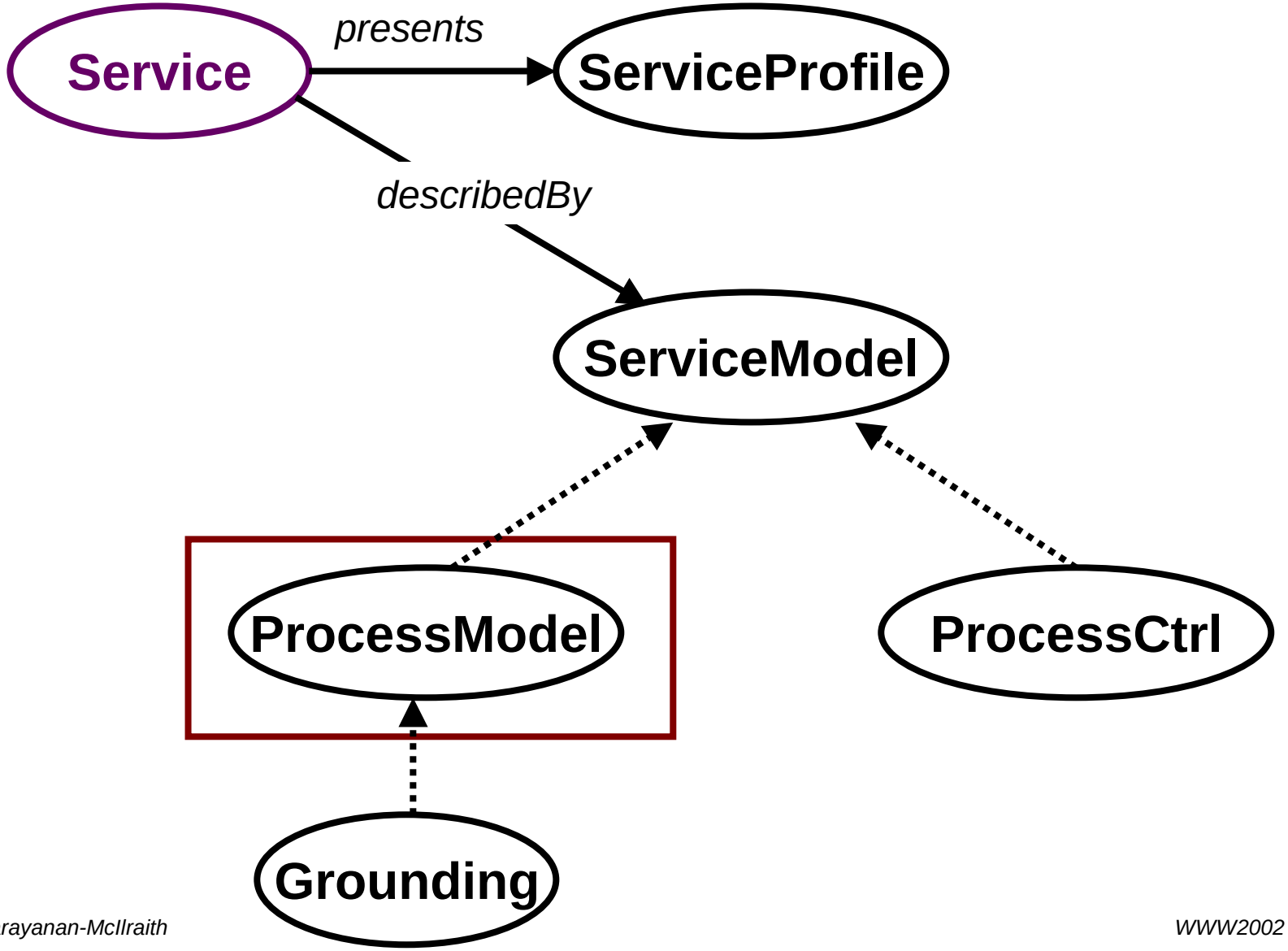
- DAML+OIL ontology for Web services:
  - well-defined semantics
  - ontologies support reuse, mapping, succinct markup, ...

- Developed by a coalition of researchers from Stanford, SRI, CMU, BBN, and Nokia, Yale, under the auspices of DARPA.

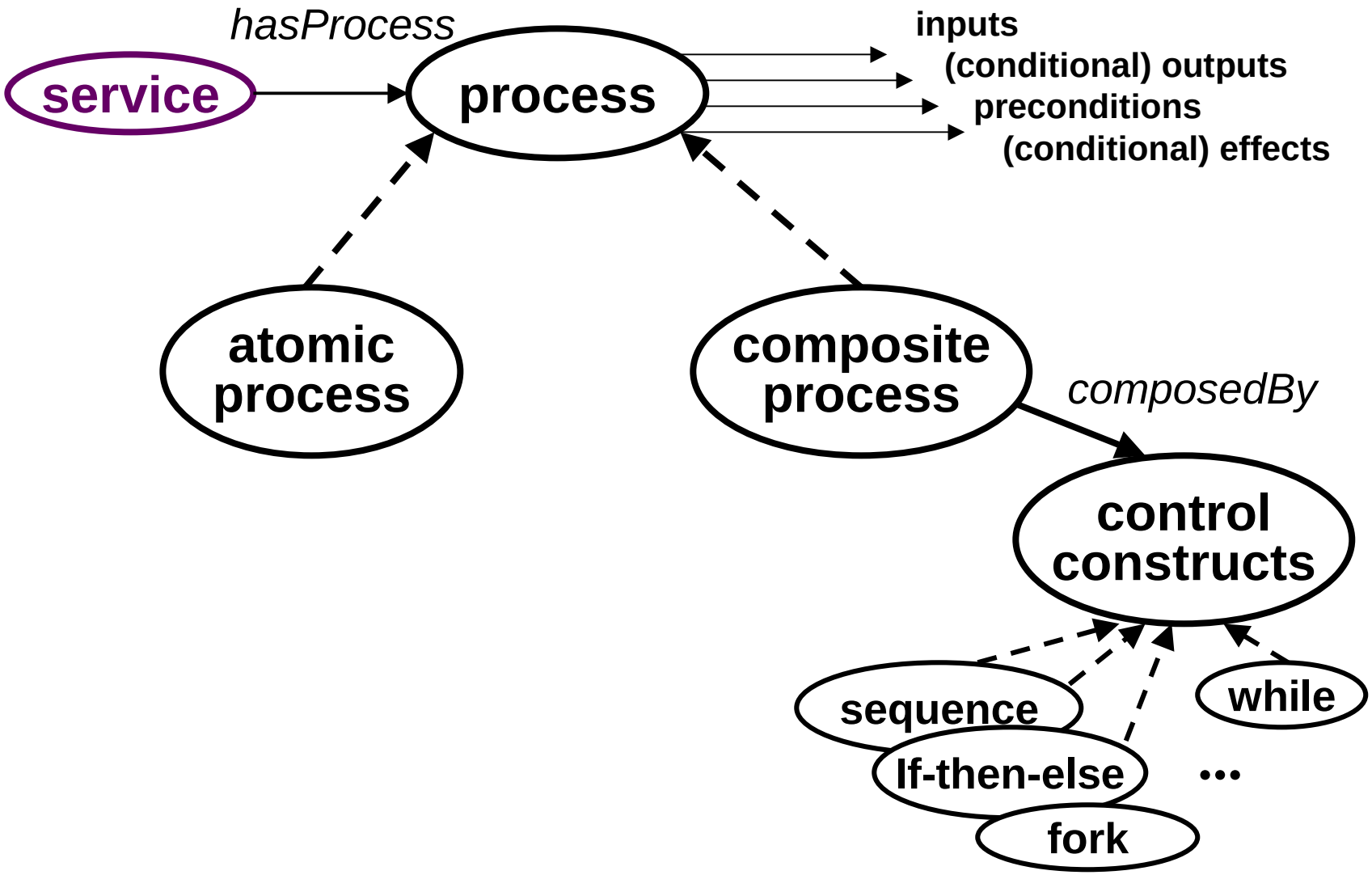
- DAML-S version 0.6 posted October,2001
  http://www.daml.org/services/daml-s

[DAML-S Coalition, 2001, 2002]
[McIlraith, Son & Zeng, 2001]

An Ontology for Service Descriptions



Process Model



Semantics of DAML-S Process Models

Problem: DAML+OIL has a well-defined semantics, but it is *not* sufficiently expressive to characterize all and only the intended interpretations of DAML-S.

Solution:

- Model-theoretic semantics defined by a translation to (mostly) first-order logic.

- Distributed operational semantics defined in terms of Petri Nets.

This Talk

- ✓ DAML-S
  - (conditional) side effects of services are critical for WS composition
  - Description logic not expressive enough for process modeling

- ➔ Model-Theoretic Semantics for DAML-S

- Distributed Operational Semantics of DAML-S

- Decision Procedures for Web Service Automation

- Implementation of DAML-S Decision Procedures

- Summary & Future Work

Model Theoretic Semantics

Task: Capture the intended interpretation of DAML-S by translating to a more expressive logic.

Approach: Translate DAML-S to Situation Calculus (SC), a first-order logical language for reasoning about action and change.

Key Idea: Preconditions = SC Preconditions
Effects = SC Effects
Inputs = SC Knowledge Preconditions
Outputs = SC Knowledge Effects

<Details of the logical translation are in the paper>

This Talk

✓ DAML-S

✓ Model-Theoretic Semantics for DAML-S

- Intended semantics of DAML-S process model via translation to FOL action theory.

➔ Distributed Operational Semantics of DAML-S

• Decision Procedures for Web Service Automation

• Implementation of DAML-S Decision Procedures

• Summary & Future Work

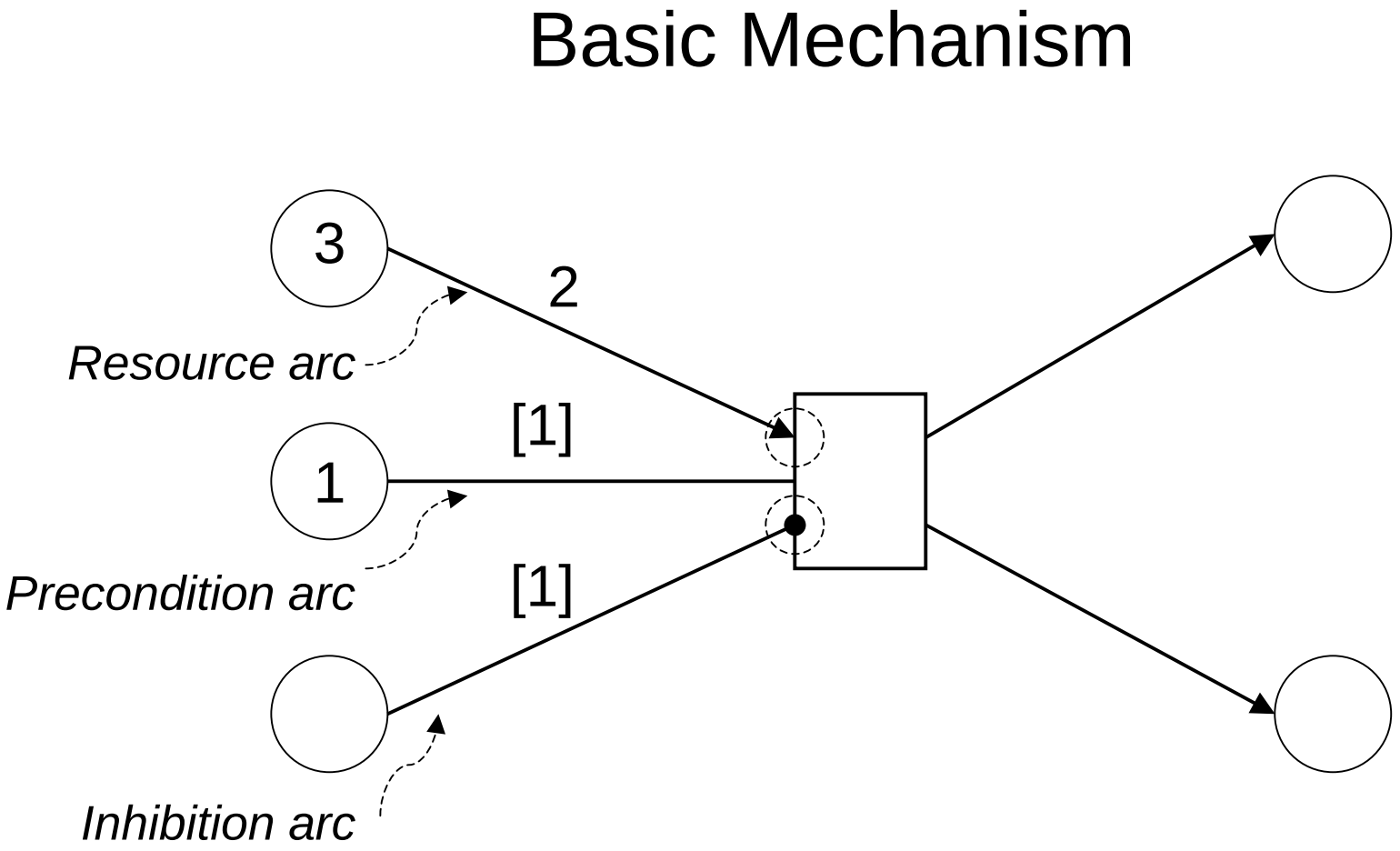
Distributed OPErational (DOPE) Semantics

Task: Map Situation Calculus Axiomatization to Petri Net based Formalism [Narayanan 99]

Features of High Level Stochastic Petri Nets

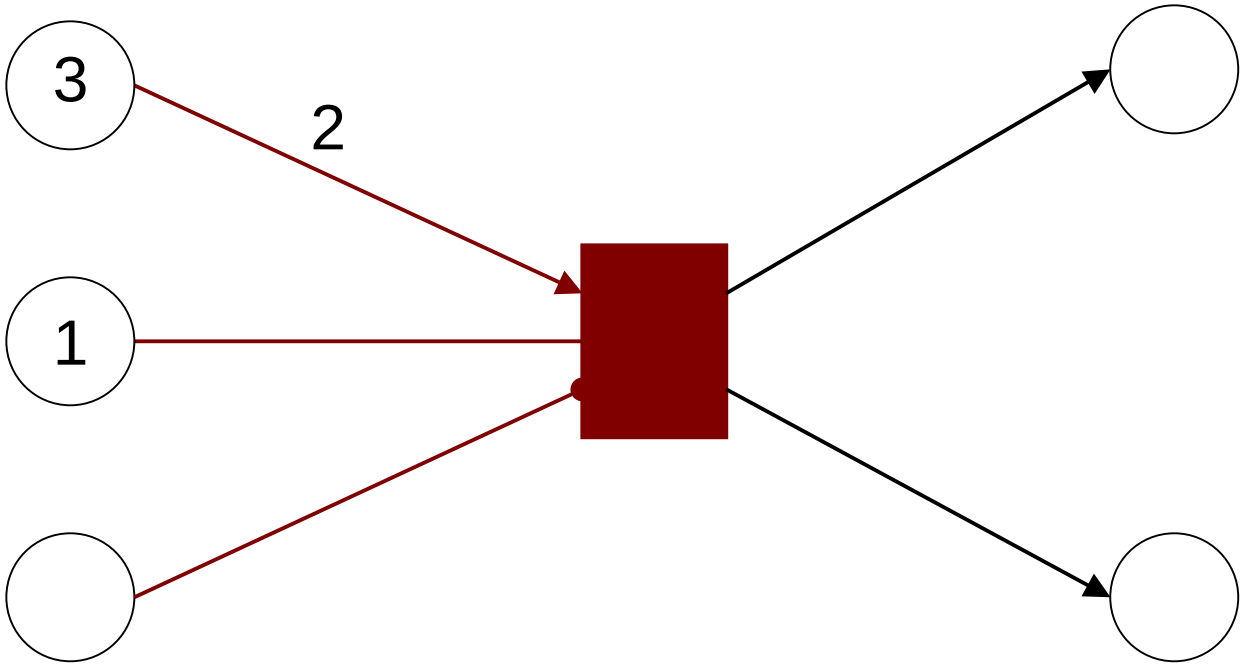
- Natural representation of change and concurrency
- Execution semantics
- Can deal with quantitative information & resources
- Variety of well established analysis and simulation techniques including mappings to other logics of change.

Model Review



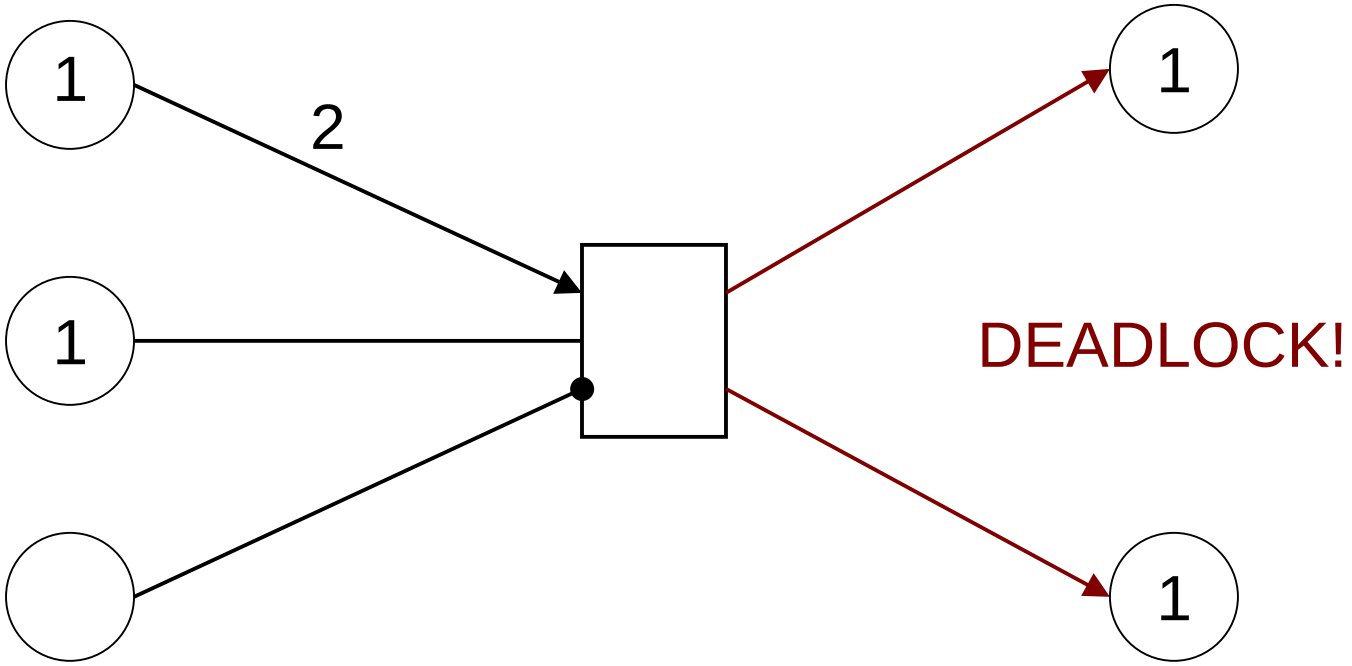
Model Review

Firing Semantics



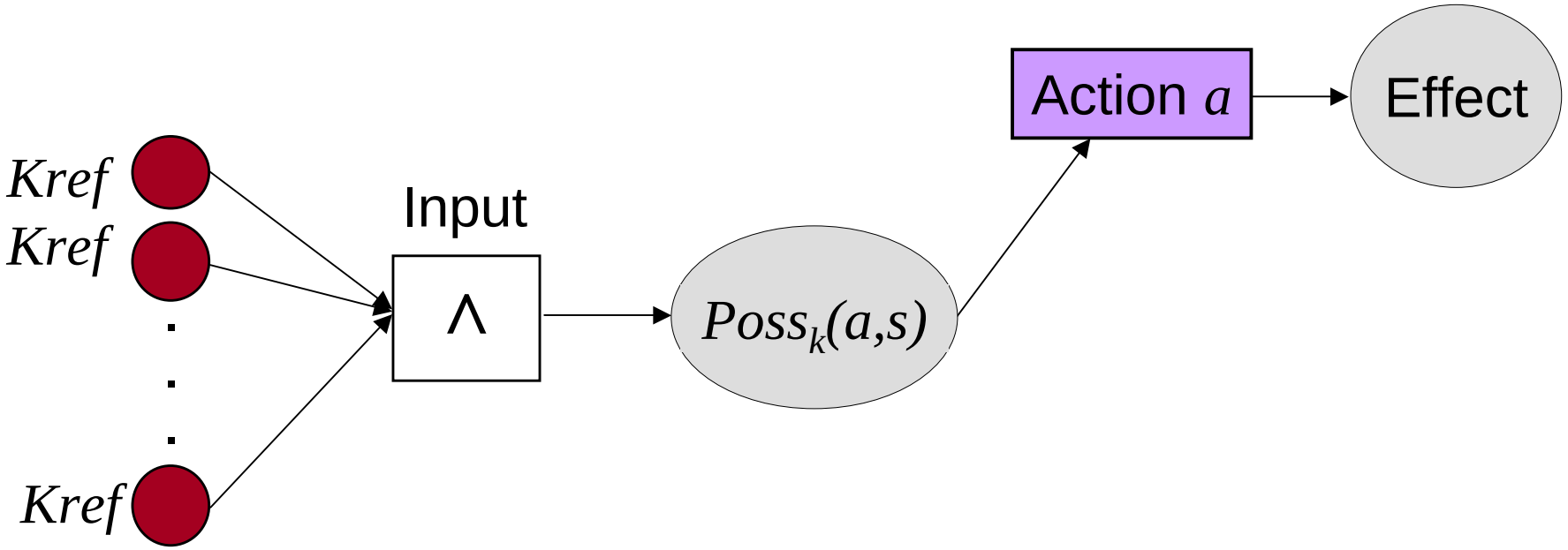
Model Review

Result of Firing



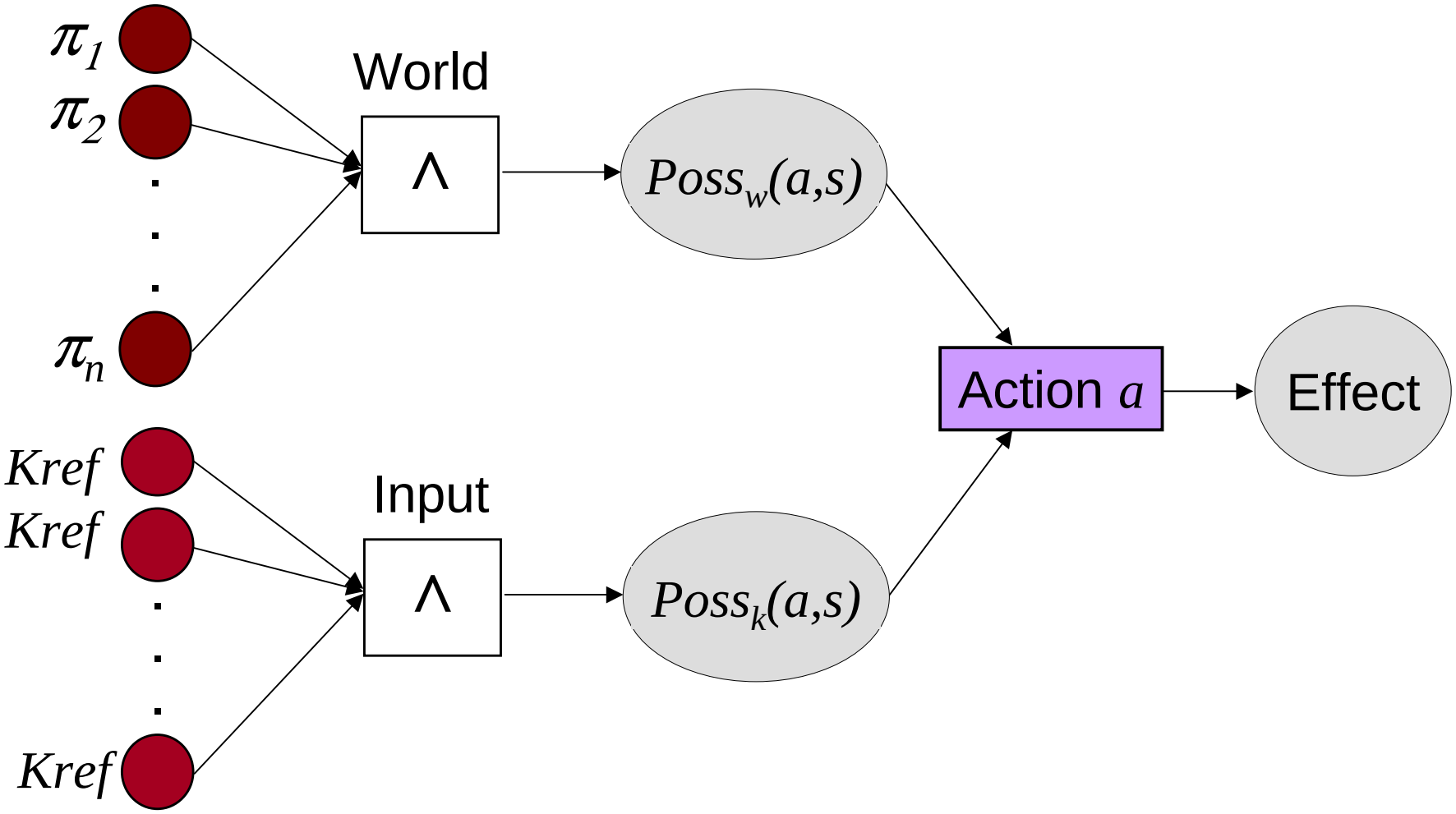
DAML-S Atomic Processes as Petri Nets

SC Knowledge Preconditions

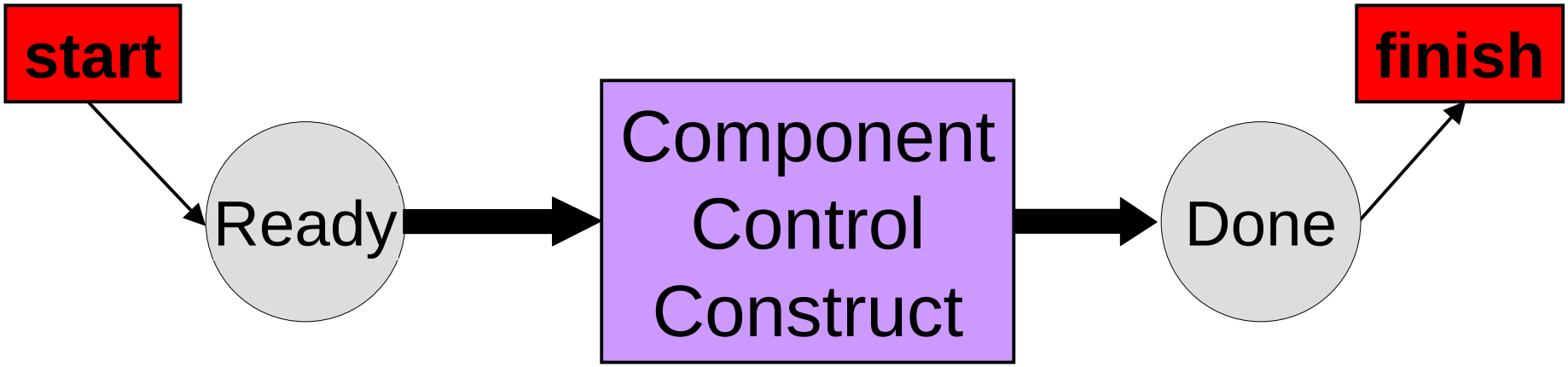


DAML-S Atomic Processes as Petri Nets

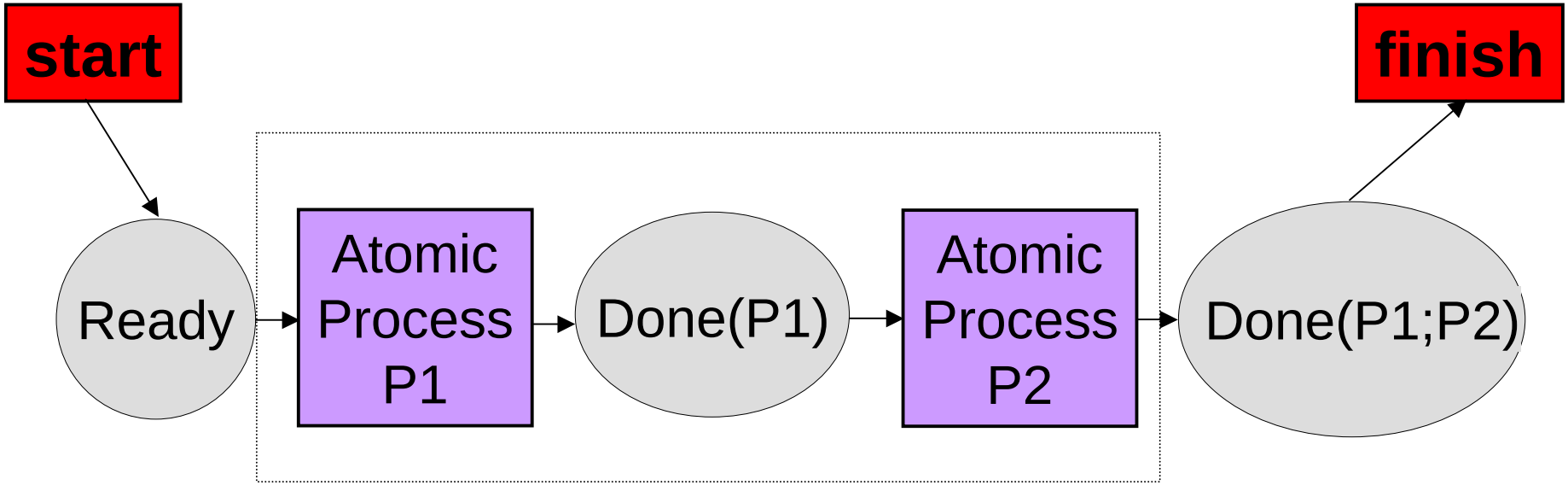
…and SC World Preconditions



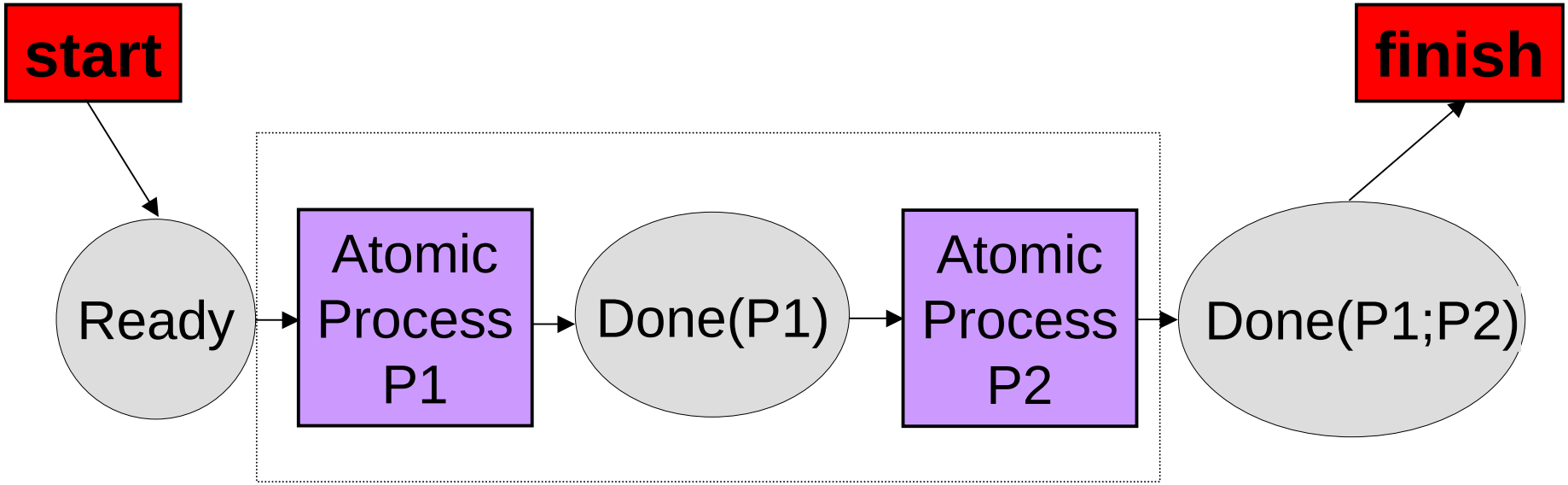
Modeling Composite Process Constructs



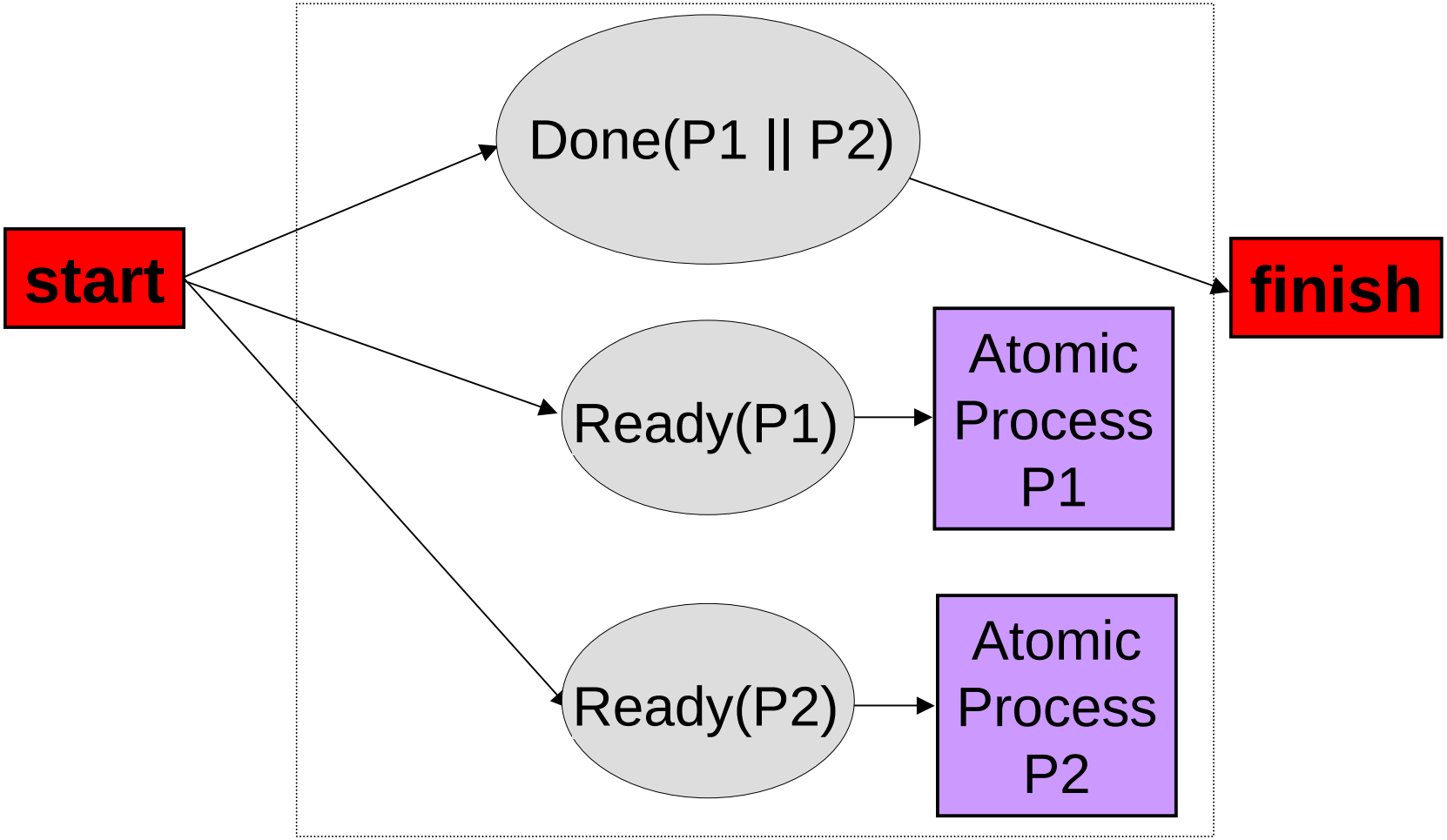
DAML-S Sequence: P1;P2



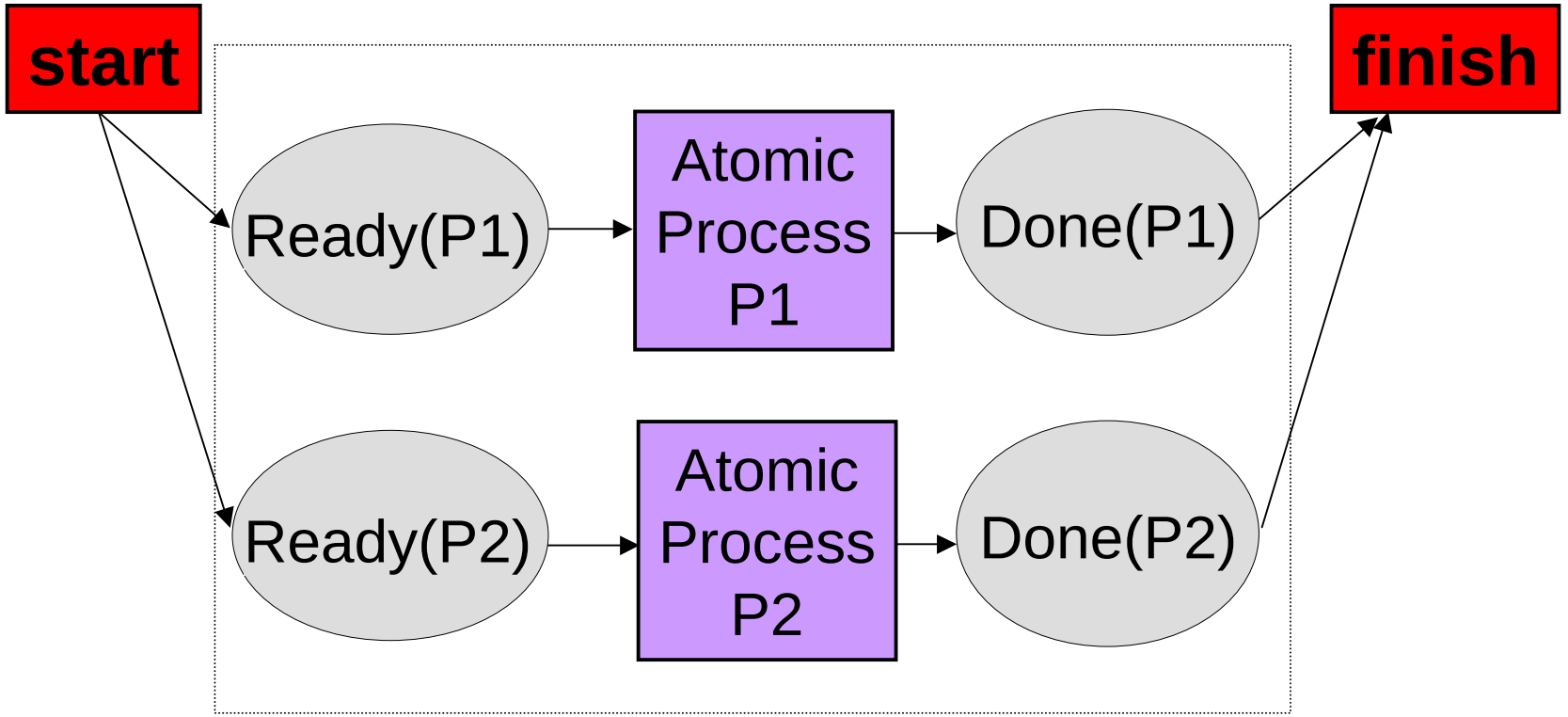
DAML-S Sequence: P1;P2



DAML-S Fork: P1|| P2



DAML-S Concurrent-Sync



This Talk

- ✓ DAML-S
- ✓ Model-Theoretic Semantics for DAML-S
- ✓ Distributed Operational Semantics of DAML-S
  - Situation Calculus translated to Petri Nets
  - Petri Net composition of atomic services
- ➔ Decision Procedures for Web Service Automation
- Implementation of DAML-S Decision Procedures
- Summary & Future Work

Web Service Automation Tasks

- **Simulation**: simulate the evolution of a Web service under different conditions.
- **Verification:** automatically establish that the Web service upholds specified properties (e.g., that it maintains certain properties, that it ensures safety, etc.)
- **Composition:** automatically generate a composition of Web services that achieves a specified goal.
- **Performance Analysis**: evaluate the ability of a service to meet requirements with respect to throughput times, service levels, and resource utilization.

Key Decision Problem 1: Reachability

Reachability: A marking M is *reachable* if it is the marking reached by some *occurrence sequence* (Definition 4, paper). Given a marking M of N, the set of reachable markings of the net (P; T; F; M) (i.e., the net obtained by replacing the initial marking M_0 by M) is denoted by [M >.

- Safety = not (reachable (unsafe state))
- Composition = reachable (goal state)

Key Decision Problem 2: Deadlock

Deadlock: A marking of a net is a *deadlock* if it enables no transitions. The deadlock problem for a net is the problem of deciding if any of its reachable markings is a deadlock.

Our Web Service Automation Task
(Simulation, Verification, Automated Composition)
can all be characterized in terms of
Reachability and Deadlock

Complexity of DAML-S Decision Procedures

Key Idea: Leverage expressiveness to gain tractability

DAML-S Subset	Reachability	Deadlock
DAML-S 0.5	P-Space hard	P-Space hard
DAML-S \ Iterate	NP-Complete	Polynomial
DAML-S \ Iterate & Cond	NP-Complete	Linear
DAML-S \ Iterate & Choice	Polynomial	Polynomial
+ Resources	Exp-Space-Time hard	Exp-Space-Time hard

This Talk

- ✓ DAML-S
- ✓ Model-Theoretic Semantics for DAML-S
- ✓ Distributed Operational Semantics of DAML-S
- ✓ Decision Procedures for Web Service Automation
  - reachability & deadlock are key for simulation, verification and automated composition of Web Services.
  - tradeoff DAML-S expressiveness for tractability of decision procs.
- ➔ Implementation of DAML-S Decision Procedures
- Summary & Future Work

Implementation

DAML-S translation to the modeling environment KarmaSIM [Narayanan, 97]
(http://www.icsi.berkeley.edu/~snarayan)

Basic Program:

Input: DAML-S description of Web Service

Output: Network Description of Web Service in KarmaSIM

Procedure:

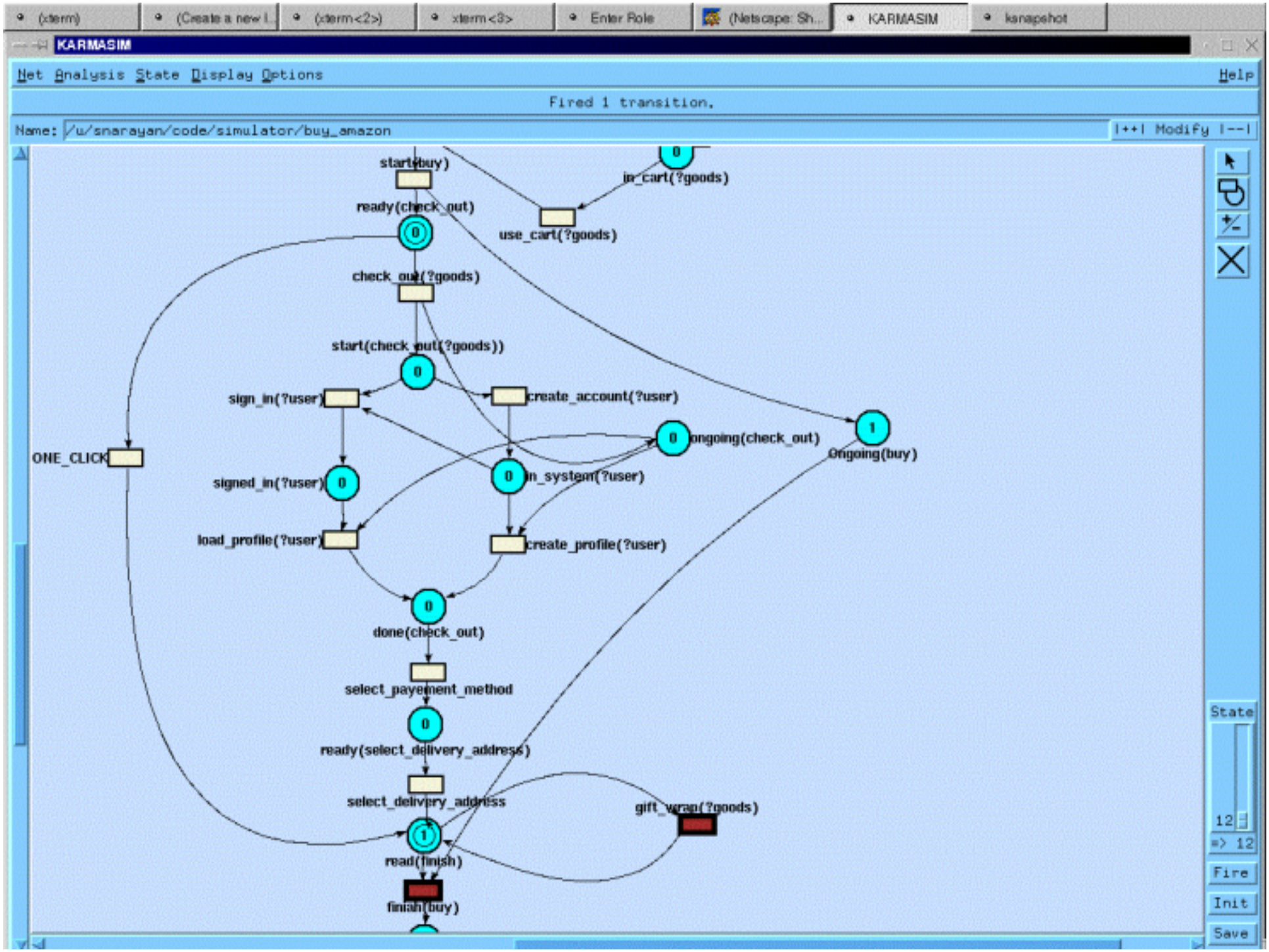
- Recursively construct a sub-network for each control construct. Bottom out at atomic process.
- Construct a net for each atomic process
- Return network

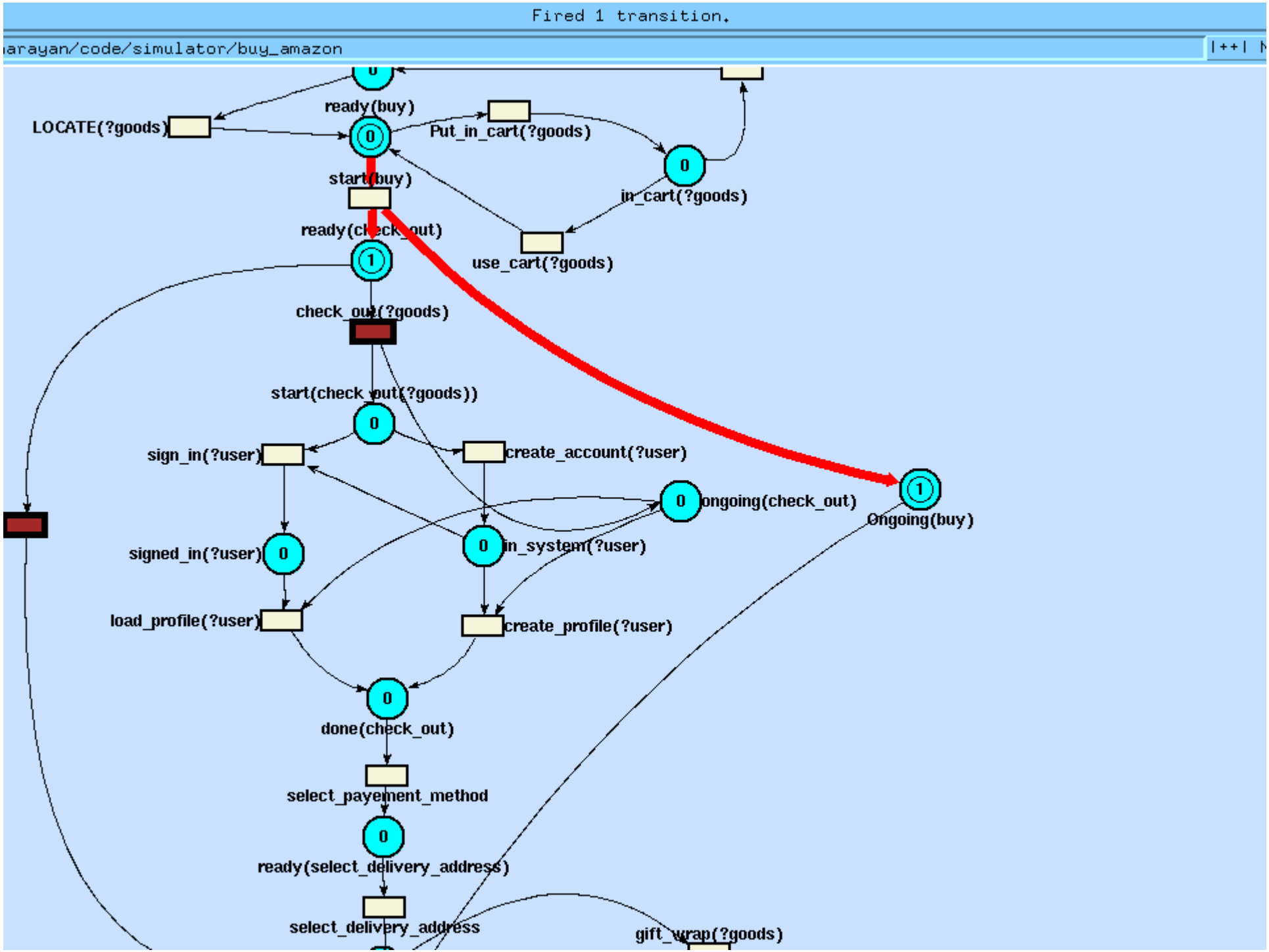
Example

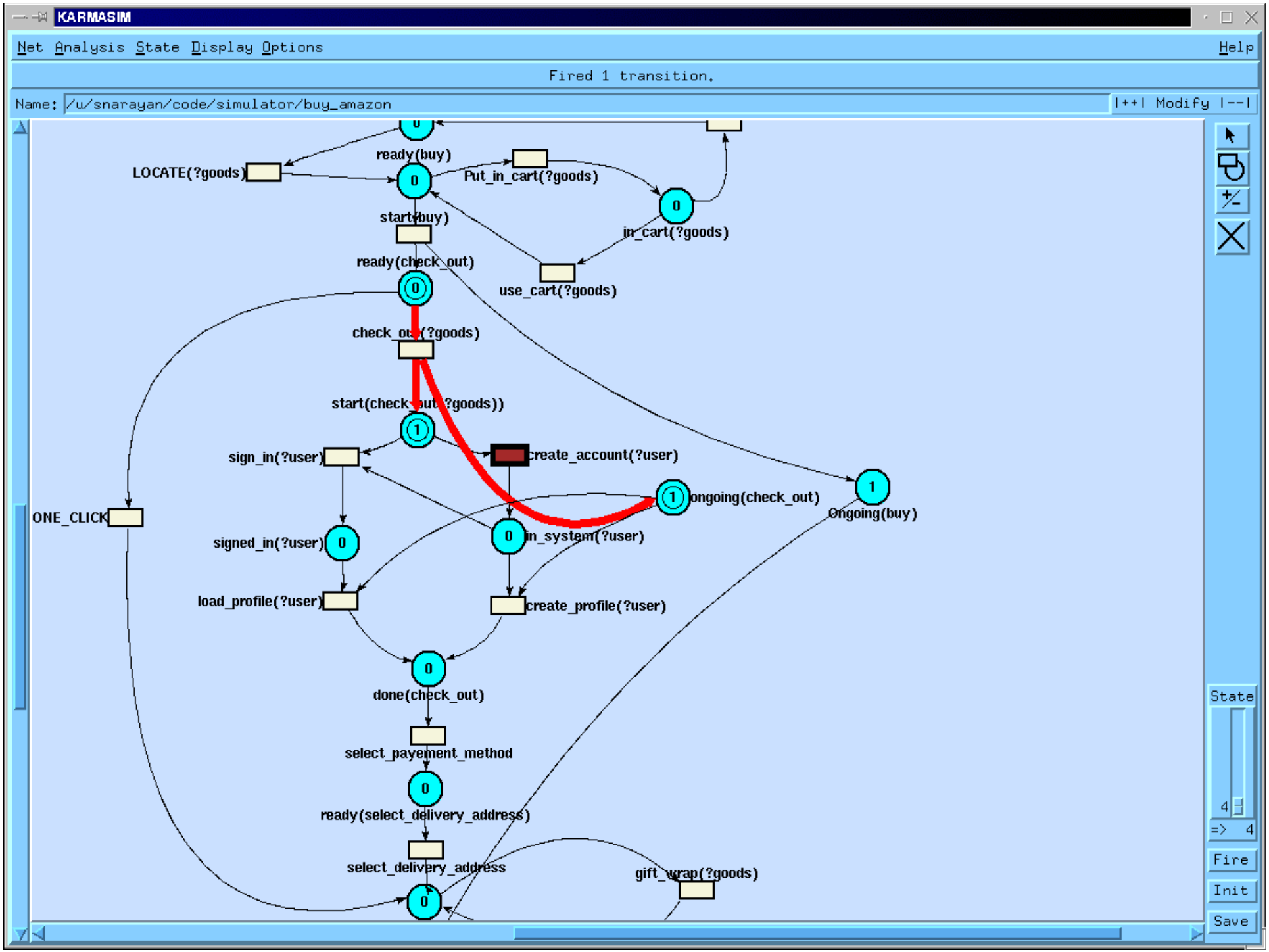
Congo.daml: A ficticious book selling service.

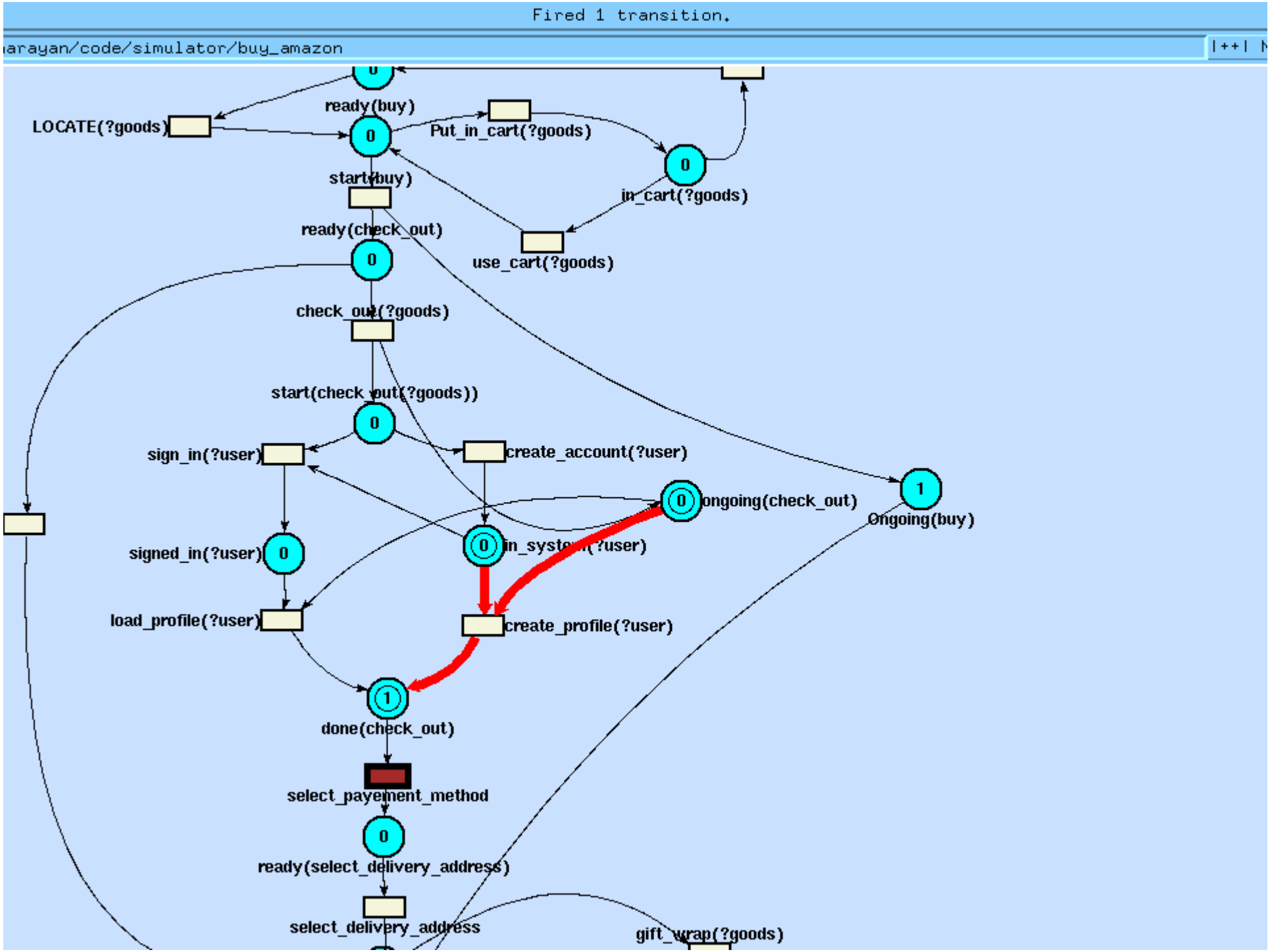
Developed by the DAML-S coalition, publically available at

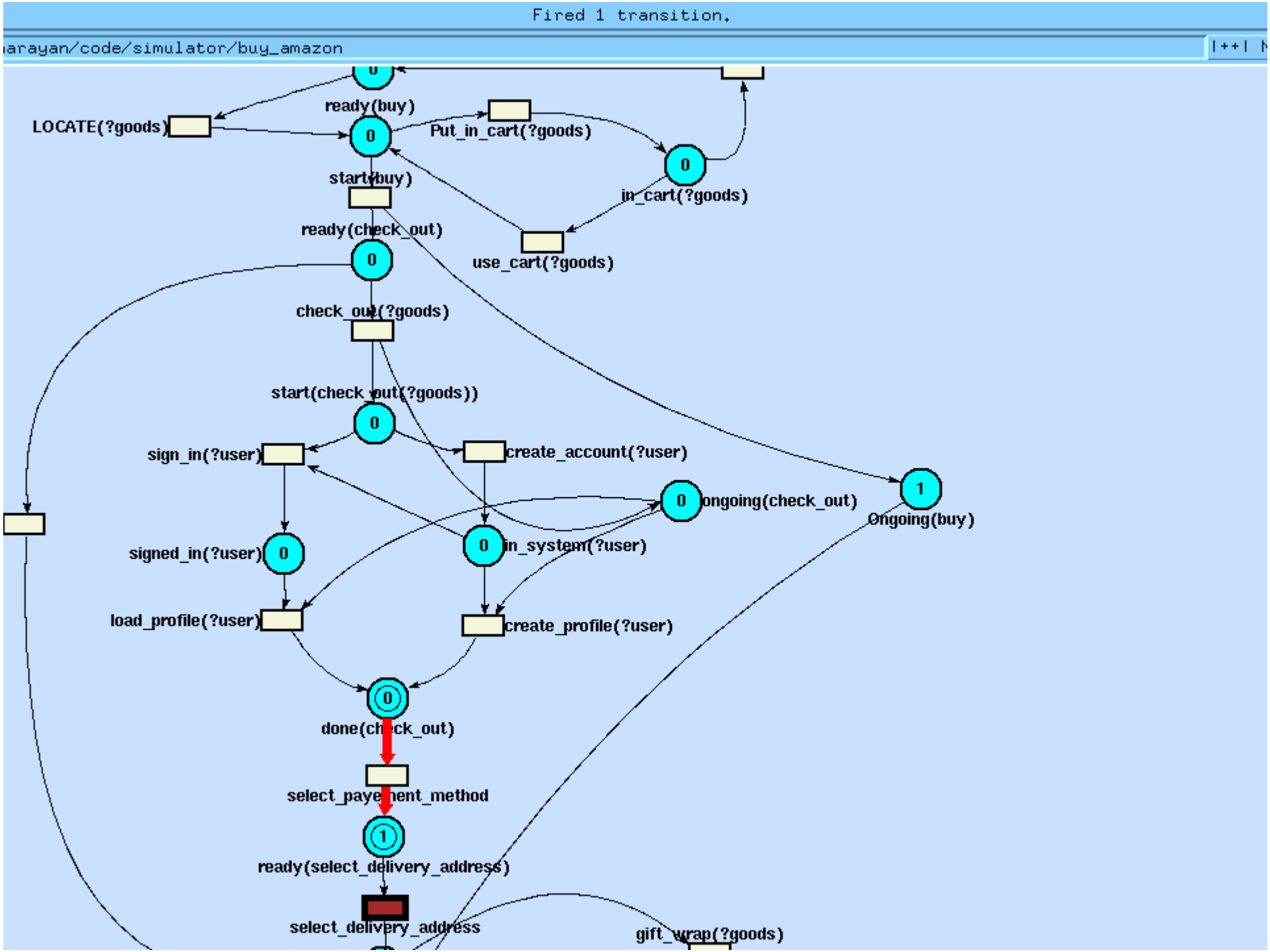
http://www.daml.org/services/daml-s/2001/05/Congo.daml

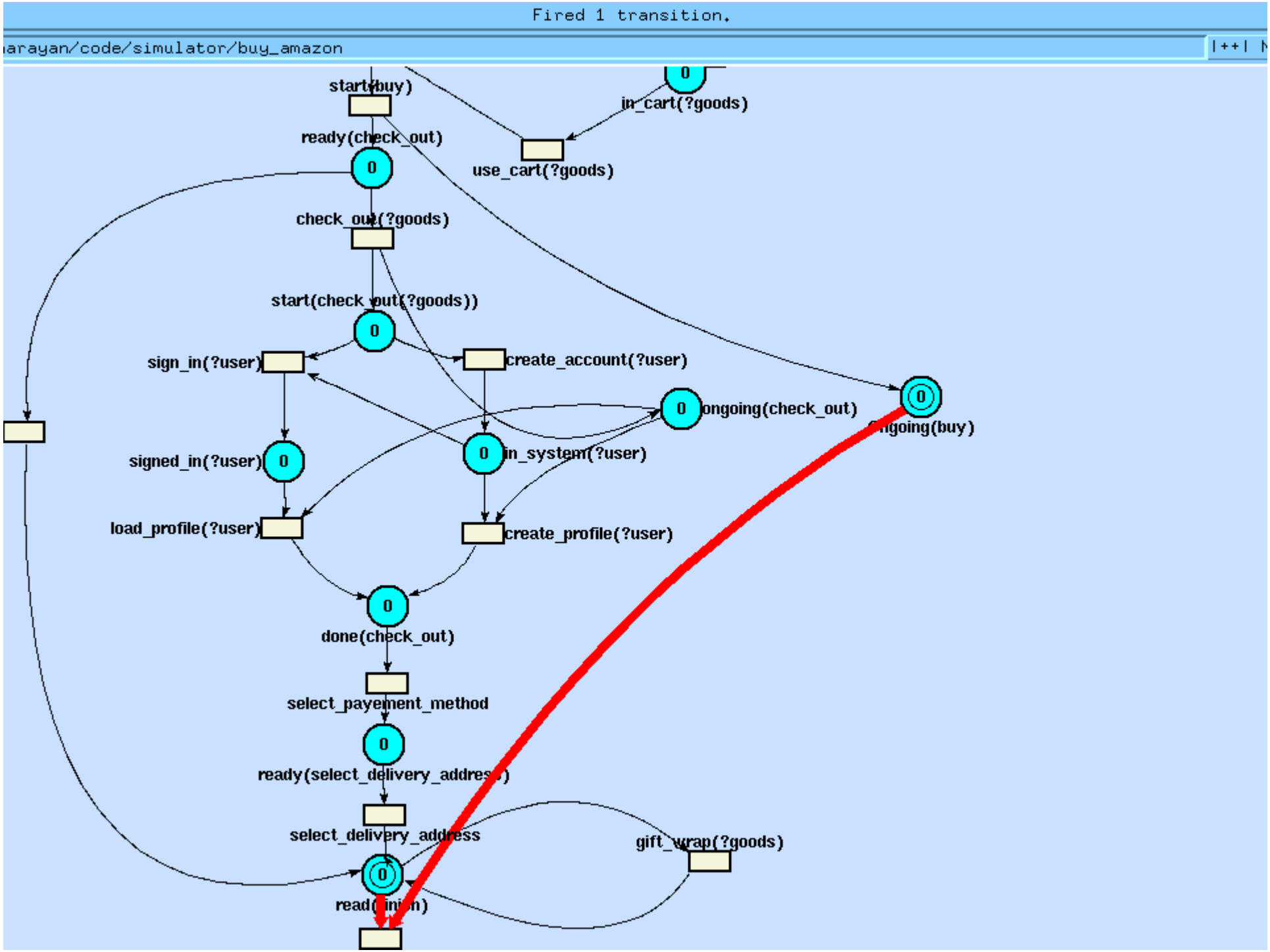

(xterm)
(Create a new l...
(xterm<2>)
xterm<3>
Enter Role
(Netscape: Sh...
KARMASIM
snapshot
KARMASIM
Net Analysis State Display Options
Help
Fired 1 transition.
Name: /u/snarayan/code/simulator/buy_amazon
|++| Modify |--|
start(buy)
in_cart(?goods)
ready(check_out)
use_cart(?goods)
check_out(?goods)
start(check_out(?goods))
sign_in(?user)
create_account(?user)
ONE_CLICK
ongoing(check_out)
Ongoing(buy)
signed_in(?user)
in_system(?user)
load_profile(?user)
create_profile(?user)
done(check_out)
select_payement_method
ready(select_delivery_address)
select_delivery_address
gift_wrap(?goods)
read(finish)
finish(buy)
State
12
=> 12
Fire
Init
Save


Fired 1 transition.
arayan/code/simulator/buy_amazon
ready(buy)
LOCATE(?goods)
Put_in_cart(?goods)
start(buy)
in_cart(?goods)
use_cart(?goods)
check_out(?goods)
start(check_out(?goods))
sign_in(?user)
create_account(?user)
ongoing(check_out)
Ongoing(buy)
signed_in(?user)
in_system(?user)
load_profile(?user)
create_profile(?user)
done(check_out)
select_payement_method
ready(select_delivery_address)
select_delivery_address
gift_wrap(?goods)


KARMASIM
Net Analysis State Display Options
Help
Fired 1 transition.
Name: /u/snarayan/code/simulator/buy_amazon
|++| Modify |--|
LOCATE(?goods)
ready(buy)
Put_in_cart(?goods)
in_cart(?goods)
start(buy)
ready(check_out)
use_cart(?goods)
check_out(?goods)
start(check_out(?goods))
sign_in(?user)
create_account(?user)
ONE_CLICK
ongoing(check_out)
Ongoing(buy)
signed_in(?user)
in_system(?user)
load_profile(?user)
create_profile(?user)
done(check_out)
select_payement_method
ready(select_delivery_address)
select_delivery_address
gift_wrap(?goods)
State
4
=> 4
Fire
Init
Save


Fired 1 transition.
arayan/code/simulator/buy_amazon
ready(buy)
LOCATE(?goods)
Put_in_cart(?goods)
in_cart(?goods)
start(buy)
ready(check_out)
use_cart(?goods)
check_out(?goods)
start(check_out(?goods))
sign_in(?user)
create_account(?user)
ongoing(check_out)
Ongoing(buy)
signed_in(?user)
load_profile(?user)
create_profile(?user)
done(check_out)
select_payement_method
ready(select_delivery_address)
select_delivery_address
gift_wrap(?goods)


Fired 1 transition.
arayan/code/simulator/buy_amazon
ready(buy)
LOCATE(?goods)
Put_in_cart(?goods)
start(buy)
in_cart(?goods)
ready(check_out)
use_cart(?goods)
check_out(?goods)
start(check_out(?goods))
sign_in(?user)
create_account(?user)
ongoing(check_out)
Ongoing(buy)
signed_in(?user)
in_system(?user)
load_profile(?user)
create_profile(?user)
done(check_out)
select_payment_method
ready(select_delivery_address)
select_delivery_address
gift_wrap(?goods)


Fired 1 transition.
arayan/code/simulator/buy_amazon
start(buy)
in_cart(?goods)
ready(check_out)
use_cart(?goods)
check_out(?goods)
start(check_out(?goods))
sign_in(?user)
create_account(?user)
ongoing(check_out)
ngoing(buy)
signed_in(?user)
in_system(?user)
load_profile(?user)
create_profile(?user)
done(check_out)
select_payement_method
ready(select_delivery_addres
select_delivery_address
gift_wrap(?goods)
read(in)

Implemented Features of Tool

- Interactive Simulation
- Variety of qualitative analysis techniques
  - Reachability
  - Deadlock
  - S and T invariants
- Variety of Quantitative analysis techniques
  - Throughput
  - MAP estimation including Vitterbi paths

This Talk

- ✓ DAML-S
- ✓ Model-Theoretic Semantics for DAML-S
- ✓ Distributed Operational Semantics of DAML-S
- ✓ Decision Procedures for Web Service Automation
- ✓ Implementation of DAML-S Decision Procedures
  - DAML-S to automated WS tasks
- ➔ Summary & Future Work

Summary

Claim

- Automation of Web Service Tasks is a key benefit of the Semantic Web.
- Precise description of Web Services is a prerequisite to Web Service automation.

Our Contributions

- Semantics (model-theoretic & distributed operational)
- Decision procedures & expressiveness-tractability tradeoff
- Implementation (from DAML-S to an executing model)

Broader Impact

- results applicable to *any* WS process model formalism (e.g., XLANG, WSFL, etc.)

Current Status/Work

- Release tool for DAML-S interpretation
  - Awaiting DAML-S stability
- Extend the Model to handle
  - Execution Monitoring
  - Resource Ontologies
- Link to Web Service Entry Tool (KSL)
- Link to general inference tool (SRI)