Towards Verification of Systems of Asynchronous Concurrent Processes

Marek Rychlý

Department of Information Systems,
Faculty of Information Technology,
Brno University of Technology

Outline

- Introduction
- Distributed information systems
- Asynchronous network model
- Process algebras (CCS, π-calculus, …)
- Modified asynchronous network model
- Framework for modified network model
- Formal specification
- Formal verification
- Future research
- References

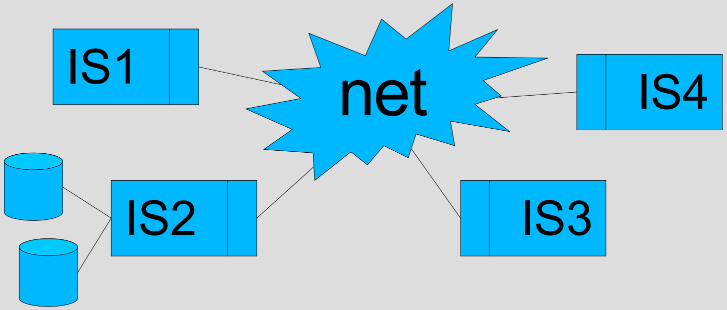
Introduction

- What will be the presentation about?
  - a design method supported by a framework
  - distributed (networked) information systems
  - an asynchronous communication
  - a network of communicating processes
  - a specification of communication architecture

- What won't be the presentation about?
  - a logic of information systems
  - process specification
  - distributed algorithms

Distributed Information Systems

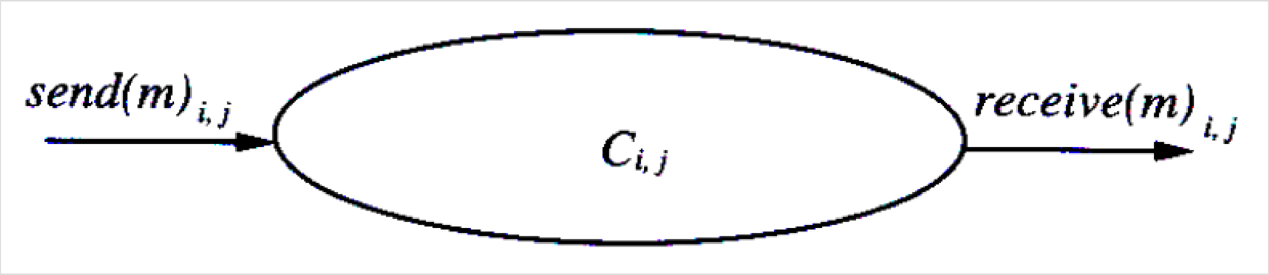
- Present-day information systems are built as SW confederations (peer-to-peer networks)
- Many autonomous components
- Gateways (interfaces) to a middle-ware
- Middle-ware provides dynamic connections
  - according to functionality (available services)
  - according to free resources
  - according to policies of components



Asynchronous network model

- directed graph of communicating processes
- edges are communicating channels
- two operations:
  - asynchronous *send(m)*$_{i,j}$
  - synchronous *receive(m)*$_{i,j}$
- many types of channels:
  - „universal reliable FIFO channel“
  - „reliable reordering channel“
  - „channel with failures“ (losses, duplications, …)

Asynchronous network model



- can be modelled as an I/O automaton
  - a labelled transition system model with output, internal and always enabled input actions and „a fair execution“
  - developed by Lynch and Tuttle, 1987

Process algebras
(„process calculus“, „process theory“)

- algebraic approach to system of concurrent processes (high level of abstraction)
- formal verification
  - synchronization (critical sections)
  - liveliness, fair execution (deadlocks)
  - temporal logics (to describe properties of executions)
- *Calculus of Communicating Systems* (CCS) Milner, 80th and 90th years
- *Communicating Sequential Processes* (CSP) Hoare, 1984-85

π-calculus
(calculus of mobile processes)

- R. Milner, J. Parrow a D. Walker (1992): *A Calculus of Mobile Processes*
- CCS with dynamic comm. structures
- only two concepts:
  - agent: communicating process,
  - name: comm. channel, variable, data, …
- key properties:
  - name passing
  - replication
- modifications:
  - polyadic, with replication, non-recursive, high-order, with name equality, …

π-calculus: operations

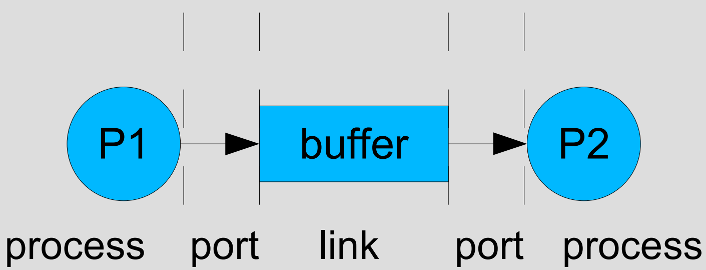
- $x<y>.P$ – operation „send“
- $x(y).P$ – operation „receive“
- $tau.P$ – internal (hidden) action
- $(x)P$ – new name
- $P|Q$ – parallel composition
- $P+Q$ – non-deterministic choice
- $A(x_1,\ldots,x_n)$ – agent execution
- $[x=y]P$ – name equality (extension)
- $!P$ – replication (extension)

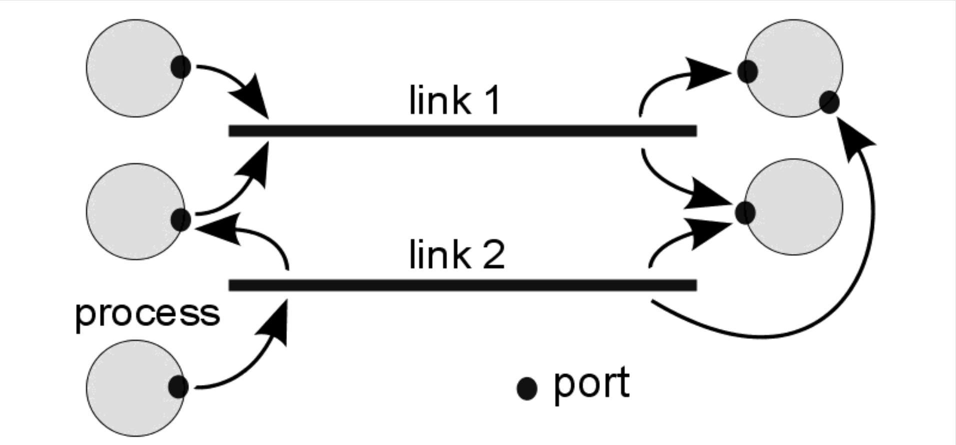
π-calculus: proofs

- Implementation of lambda-calculus (Robin Milner, 1992)
- Bisimulation equivalence:
  - early and late: input action after/before substitution (isn't congruent, Milner 1992)
  - open bisimulation: all actions (is congruent, Sangiorgy 1996)
- Proof of bisimulation equivalence in finite recursive π-calculus (Mads Dam, 1997)
  - auto-prover (Björn a Moller, 1994) The Mobility Workbench - A Tool for the π-Calculus

Modified asynchronous network model

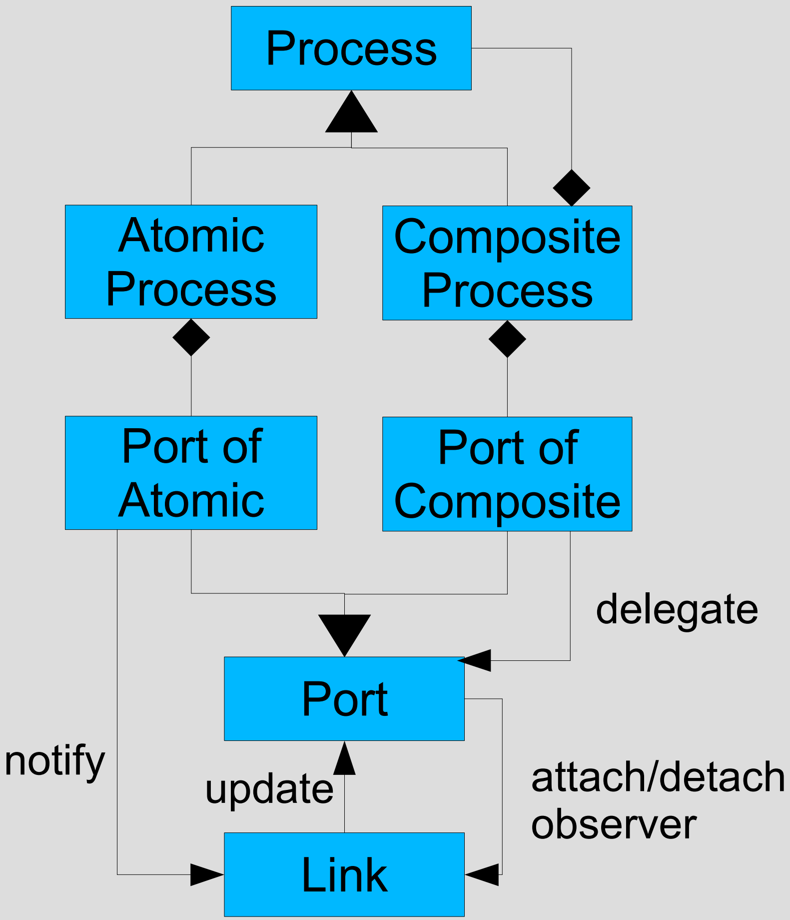
- Original ANM: process and channel
- Decomposition of channel:
  - local interface … „port“
  - network buffer … „link“
- Translatable into original ANM





Framework for modified asynchronous network model

- Tool for modelling in modified network model
- *white-box* framework (and implementation library of components for *black-box* fmw.)
- Hierarchy and encapsulation of processes



Formal specification

- High level of abstraction in the model
  - focused on the communication
  - unknown semantics of atomic processes
- Systems implemented using the framework are compatible with modified network model
- Systems implemented using the framework can be translated into π-calculus
- We suppose „universal reliable FIFO channel“ in formal specification (ideal)

Specification in π-calculus

- The atomic process is process of π-calculus
- The port is two channels (input and output)
- The link is expressed as π-calculus process, which connects input and output channels:

$$\text{link}(p1_{in}, \ldots, pn_{in}, q1_{out}, \ldots, qm_{out}) = \sum_{i=1}^{n} \sum_{j=1}^{m} qj_{out}(x).\overline{pi_{in}}\, x\,.\text{link}(p1_{in}, \ldots, pn_{in}, q1_{out}, \ldots, qm_{out})$$

- The composite process is a parametric process (a parallel composition of its internal processes) with the ports of a composite process as its parameters

Formal verification

- After translation into π-calculus in MWB
- Problem with infinite recursion (replication)
  - Can be replaced with a finite number of concurrent processes?
  - Is it possible to use some recycling mechanism?
- We can:
  - prove weak and strong open bisimulation equiv.
  - find deadlocks
  - simulate and test system (as „a black-box“ and „a white-box“)

Future research

- Model:
  - Elimination of an infinite recursion
  - Influence of a network layer QoS on the model
  - Relation with UML2 (design pattern Port)
- Framework:
  - Lesser dependence on the network model
  - Framework implementation and case-studies
  - Specification of SOA, CORBA Event Service, …

References

(1) Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers. San Francisco, CA, USA. 1996.

(2) Robin Milner, Joachim Parrow, and David J. Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–40 and 41–77, 1992.

(3) Victor Björn and Faron Moller. The Mobility Workbench — a tool for the π-calculus. In David Dill, editor, *CAV'94: Computer Aided Verification*, volume 818 of *Lecture Notes in Computer Science*, pages 428–440. Springer-Verlag, 1994.

(4) Ugo Montanari and Marco Pistore. Finite state verification for the asynchronous π-calculus. In *TACAS '99: Proceedings of the 5th International Conference on Tools and Algorithms for Construction and Analysis of Systems*, pages 255–269, London, UK. Springer-Verlag, 1999.

(5) Mads Dam. Proof systems for π-calculus logics. In R. de Queiroz, editor, *Logic for Concurrency and Synchronisation*, Trends in Logic, Studia Logica Library, pages 145–212. Kluwer, 2003.