Distributed Information System as a System of Asynchronous Concurrent Processes

Marek Rychlý Jaroslav Zendulka

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

2^{nd} Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, 2006

Outline

Outline

Distributed Information System

- Information systems (ISs) as distributed systems are collections of software components, communicate and coordinate their actions via a middle-ware.
- The middle-ware can provide dynamic connections, e.g.
  - according to functionality (available services),
  - according to free resources,
  - according to policies of individual components, etc.
- A component of distributed IS can be also another IS (a well-established IS).

Component-Based Software Engineering

- Software applications are assembled from components from a variety of sources.
- Different implementations but only one “black-box” specification of a component.
  - different programming languages, platforms, environments,
  - components aren’t objects – independent of implementation,
  - components aren’t services – one implementation can acts as many services.
- Maximal reusability – hierarchical composition of components.
- Formal specification of component-based architecture?

Architecture Description Languages

- Formal languages for conceptual specification of software architectures.
- Based on some formal model (Petri-nets, temporal logics, process calculus, etc.).
- Components, connectors, configurations, architecture patterns.
- Some problems:
  - domain specific design and implementation of an ADL,
  - formal specification is very difficult, even for simple systems,
  - support only static or “simple” dynamic architectures.

Dynamic Architecture

- Runtime modification of architecture:
  - creation and destruction of components and connectors,
  - passing of components, connectors as ordinary messages,
  - dynamic updating (of implementation) of components.
- How to control runtime reconfiguration and reflect it at a design time?

Outline

Motivation

Basic idea – a framework

Runtime support for formal design and implementation of an IS with dynamic component-based architecture.

- The framework will act a middle-ware providing connectors and component management.
- It'll support a hierarchy of components (atomic and composite components).
- Connectors will be independent of a transport mechanism (SOA, IPC, etc.).
- The framework will be able to control runtime reconfiguration and derive formal description of architecture.

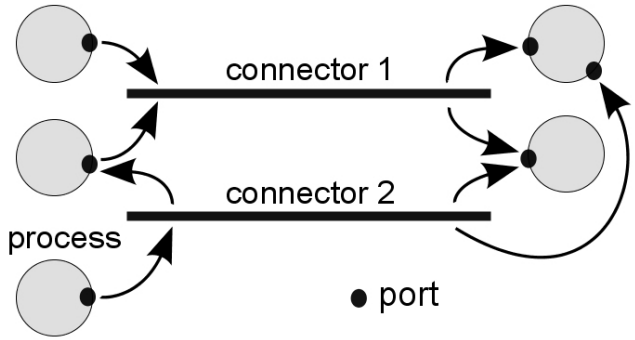
Outline

Vertical View

Modified Asynchronous Network Model (Modified ANM)

- Original ANM = directed graph of processes (nodes) communicating via channels (edges).
- Modified ANM = directed graph (compatible with the original ANM) where
  - **nodes** are processes and connectors (processes, which only resend messages),
  - **edges** are connections between processes and connectors and vice versa.

Modified Asynchronous Network Model – 3 Layers
(Vertical View)



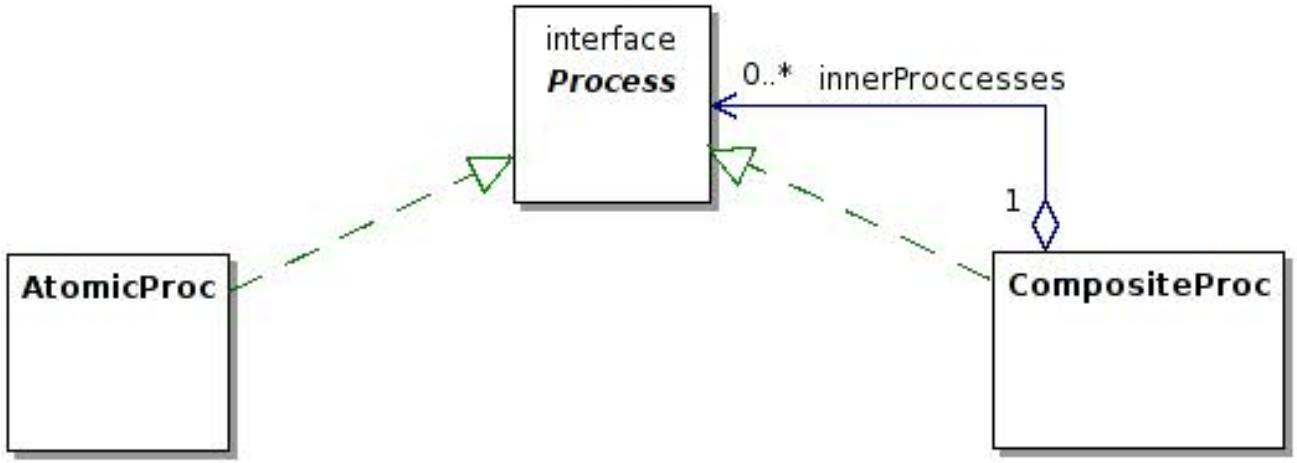
process layer – atomic or composite components, responsible for an application logic,

connector layer – connectors, a low-level communication support responsible for a reliable communication,

port layer – interfaces between the process and connector layer.

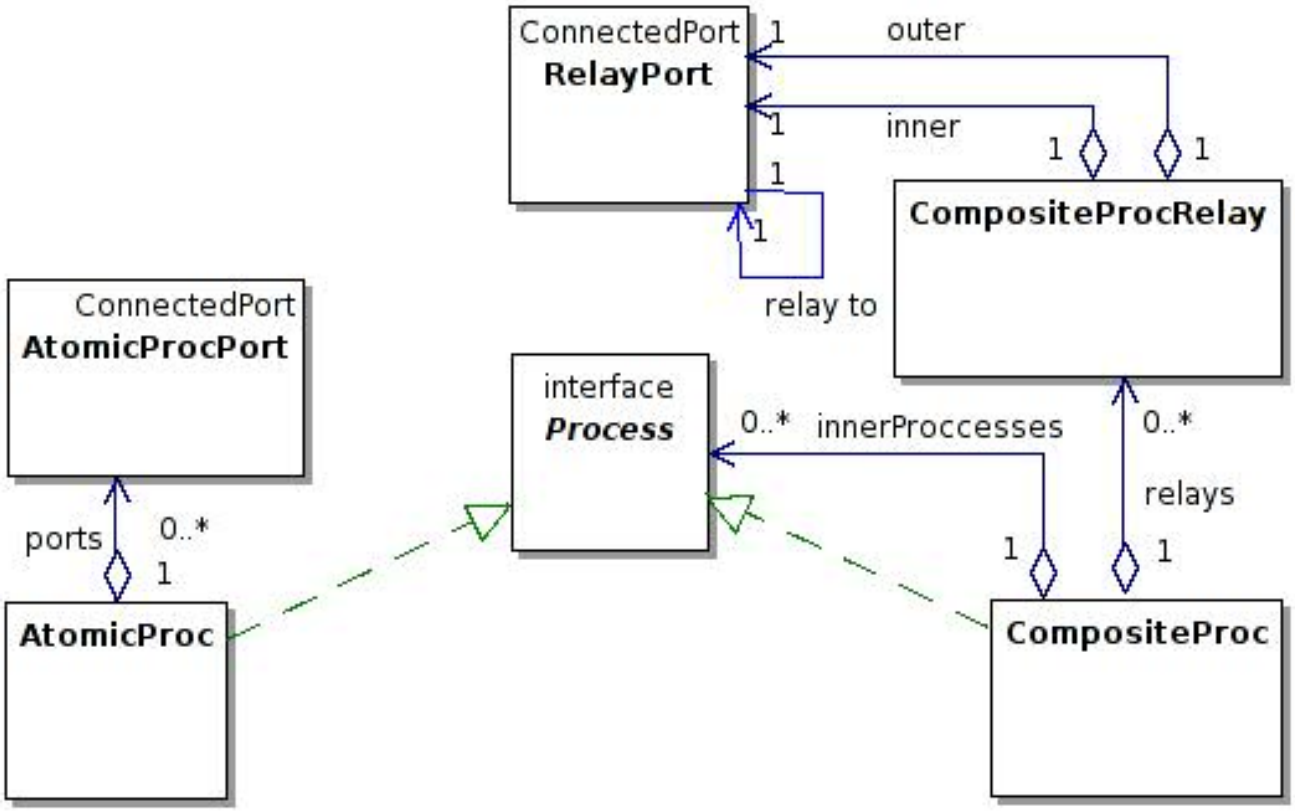
Horizontal View

Component-Based Architecture (Atomic and Composite Components)



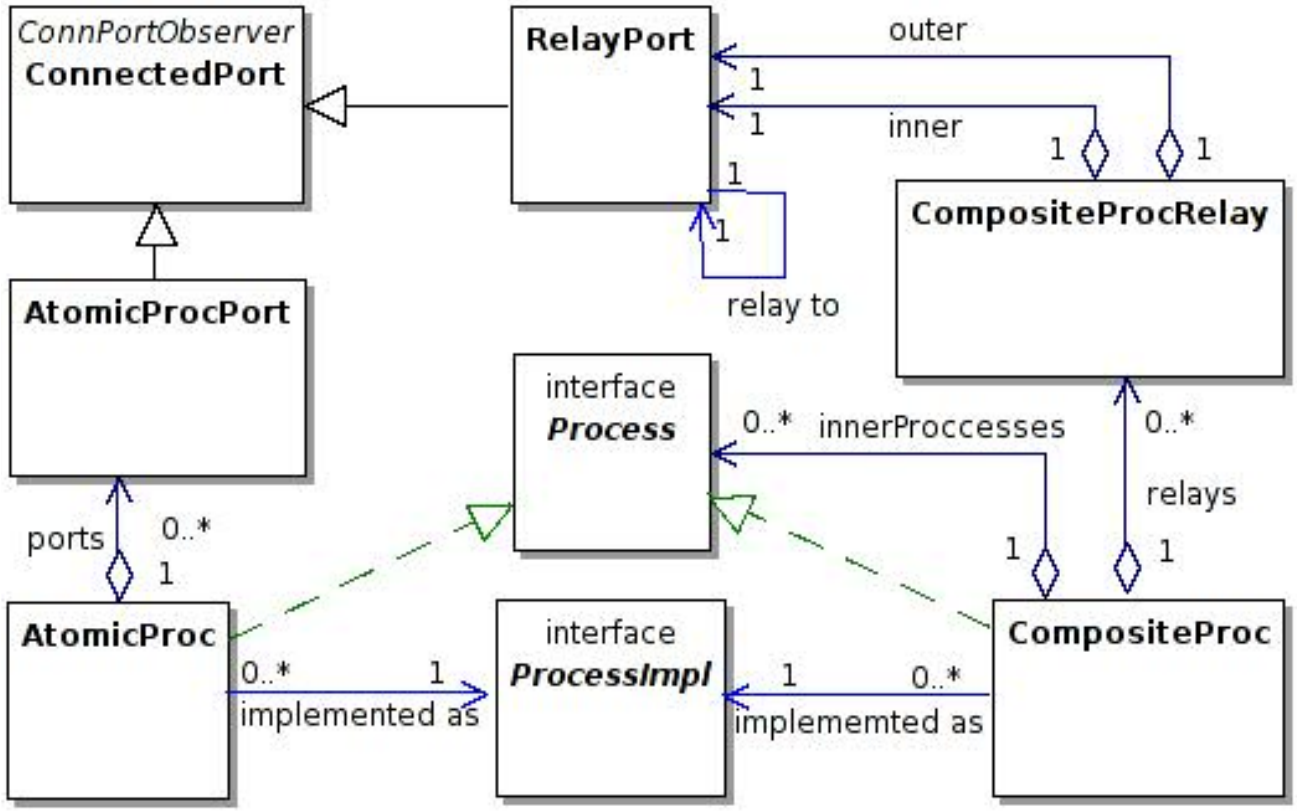
Horizontal View

Component-Based Architecture (Atomic and Composite Components)



Horizontal View

Component-Based Architecture (Atomic and Composite Components)



A Calculus of Mobile Processes (π-Calculus)

Introduction

- In 1992 by R. Milner, J. Parrow and D. Walker as modification of CCS.
- Algebraic approach to a system of concurrent and mobile processes.
- Only two concepts:

  agents – communicating processes,
  names – channels, data, (processes), etc.
- Key feature is passing of names – passing of parts of architecture.

A Calculus of Mobile Processes (π-Calculus)

Behavioural Model of a System with Dynamic Architecture

atomic components – π-calculus processes defined at a design time, interfaces of components are channels from/to the processes.

connectors – π-calculus processes with channels for connected components defined as

$$\begin{aligned} &Connector(p1_{in}, \ldots, pn_{in}, q1_{out}, \ldots, qn_{out}) = \\ &\quad \sum_{i=1}^{n} \sum_{j=1}^{m} qj_{out}(x).\overline{pi_{in}}\langle x \rangle. \\ &\quad Connector(p1_{in}, \ldots, pn_{in}, q1_{out}, \ldots, qn_{out}) \end{aligned}$$

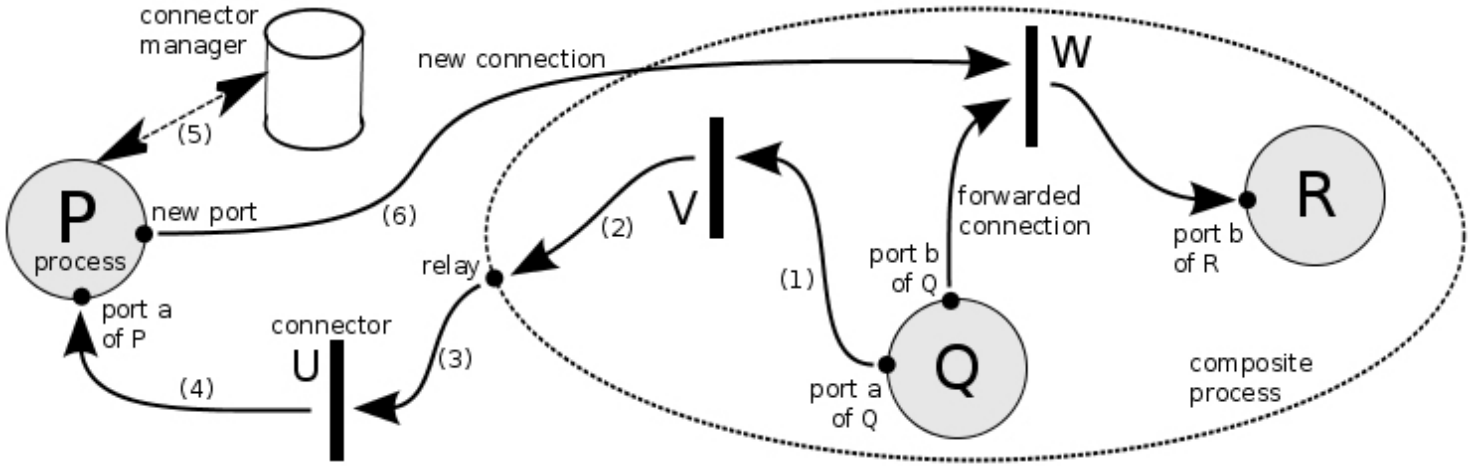
composite components – parametric π-calculus process (parallel compositions of internal processes) with hidden internal channels and visible channels of interfaces of the composite components.

Passing of Components and Connections
In the π-Calculus

- Dynamic architecture allows passing of components and connections (ports) as ordinary messages.
- In the π-calculus, it is trivial (see passing of port *b*, in the paper).
- In the framework, passing of connection to an interface of a component (a port)
  - = passing of reference to a connector, which is connected to the interface.
- After passing of a port, its environment has to be restored
  - = passed port is connected to the same connector as original port.

Passing of Components and Connections

Implementation in the Framework



- Passing of components is implemented similarly, but after the passing, environment of each port of the component has to be restored.

Summary

- Distributed ISs create needs for component-based design with dynamic architecture.
- It is a problem to control runtime reconfiguration and reflect it at a design time.
- The presented framework provides runtime support for component-based systems with dynamic architecture.

- Outlook
  - Implementation of a prototype of the framework.
  - Case-study and its evaluation.
  - Final design and implementation.

For Further Reading I


Lynch, N.A.:
Distributed Algorithms.
Morgan Kaufmann (1996)


Rychlý, M.:
Towards verification of systems of asynchronous concurrent processes.
In: *Proceedings of 9th International Conference ISIM'06*. (2006) 123–130


Milner, R., Parrow, J., Walker, D.:
A calculus of mobile processes, part I/II
Journal of Information and Computation, 100 (1992) 41–77