

Service Oriented Architectural Design

R. Bruni¹, A. Lluch Lafuente¹, U. Montanari¹, E. Tuosto²



Department of Computer Science, University of Pisa
{bruni,lafuente,ugo}@di.unipi.it

Department of Computer Science, University of Leicester
et52@mcs.le.ac.uk

3rd International Symposium on Trustworthy Global Computing
INRIA Sophia-Antipolis, November 5-6, 2007

Outline

- 1** Introduction
- 2 Architectural Design Rewriting
- 3 Design and Reconfiguration for a Service Modelling Language
- 4 The end

Our world



Principles of ADR (Architectural Design Rewriting)

- Software Architectures
 - **Designs**: graphs with interfaces.
 - Partial designs: designs with holes.
- Architectural styles
 - Set of **design productions** (operations over designs).
 - Inspired by context-free graph grammars (Le Métayer et al.).
 - A **design term** with type T is conformant to style T .
- Reconfigurations
 - Rewrite rules over design terms, not over graphs.
 - Hierarchical, inductive rules (term rewriting + SOS).
 - **Style preservation** immediate with rule $d : T \rightarrow d' : T$.
 - **No theorem or (non-terminating) algorithm needed!**

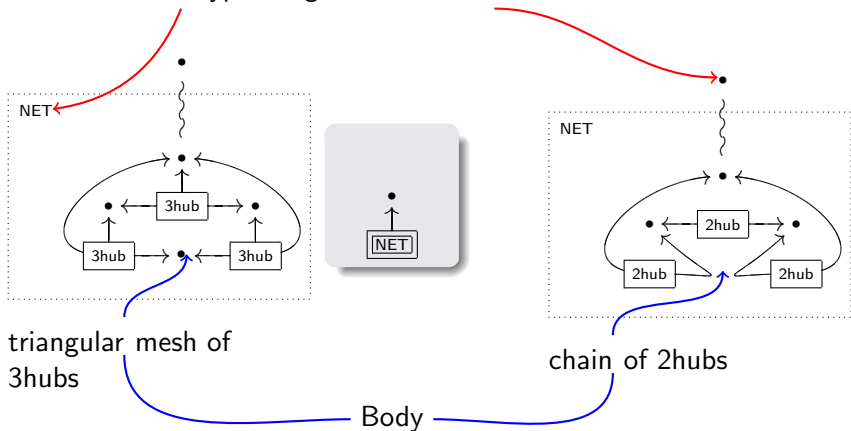
Outline

- 1 Introduction
- 2 Architectural Design Rewriting**
- 3 Design and Reconfiguration for a Service Modelling Language
- 4 The end

Scenario: Network of Chains or Triangular Meshes

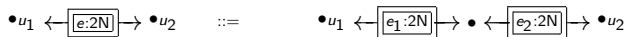
Network (design of type NET) = interface + body.

Interface = NET-typed edge with one node.

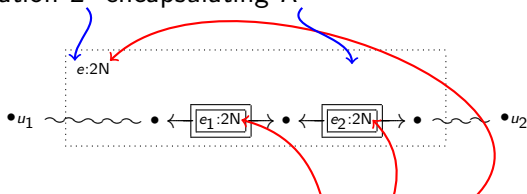


The notation of design productions

- Traditional production rule notation $L ::= R$



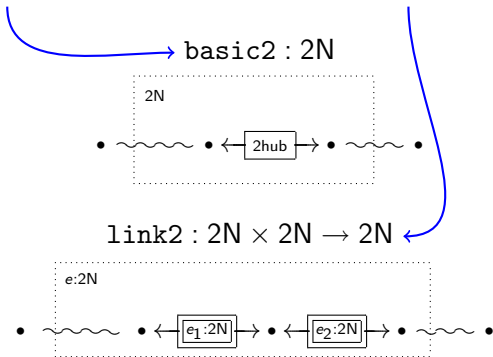
- ADR notation L encapsulating R



- ADR functional reading $\text{link2} : 2N \times 2N \rightarrow 2N$

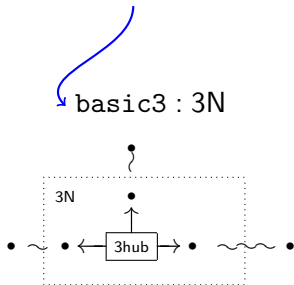
Chain style 2N

A single 2hub ... or ... two concatenated 2Ns .

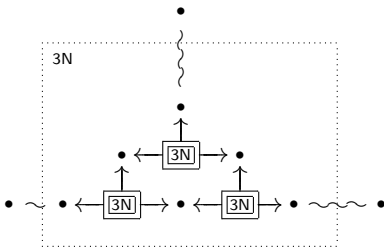


Triangular mesh style

A single 3hub ... or ... a mesh of 3hubs



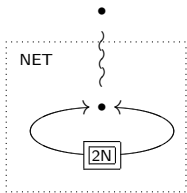
link3 : $3N \times 3N \times 3N \rightarrow 3N$



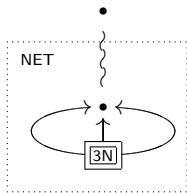
Constructing NETs

A closed chain ... or ... triangular mesh

net2 : 2N → NET

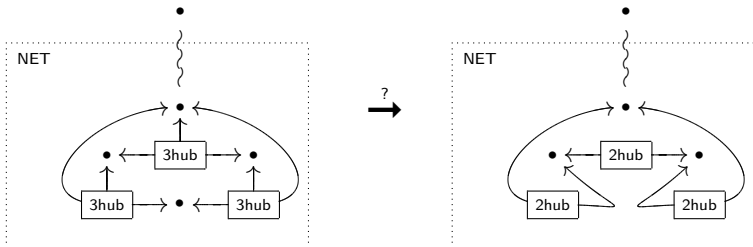


net3 : 3N → NET



Reconfiguring NETs

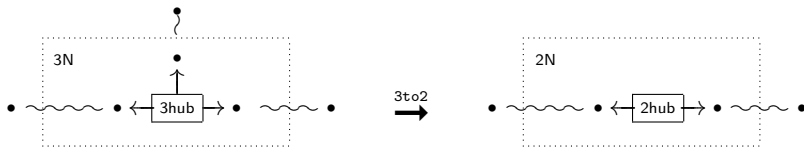
How can we transform **any** triangular NET into a chained one?



A 3hub becomes a 2hub

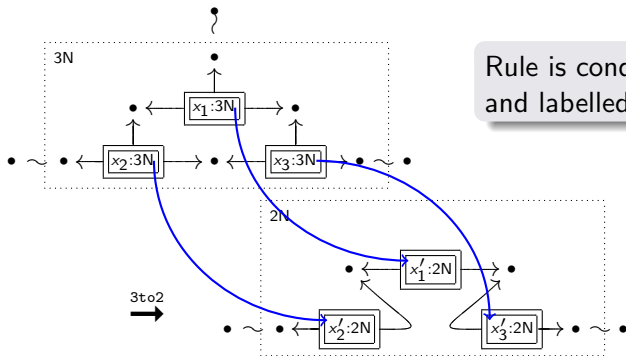
basic3 $\xrightarrow{3to2}$ basic2

Type changes (3N to 2N),
but label 3to2 takes care of contexts.



A 3N becomes a 2N

$$\frac{x_1 \xrightarrow{3to2} x'_1 \quad x_2 \xrightarrow{3to2} x'_2 \quad x_3 \xrightarrow{3to2} x'_3}{\text{link3}(x_1, x_2, x_3) \xrightarrow{3to2} \text{link2}(\text{link2}(x'_2, x'_1), x'_3)}$$

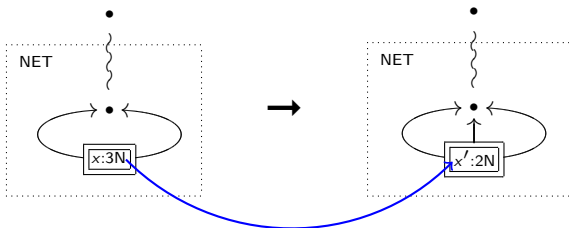


Rule is conditional
and labelled.

A triangular meshed NET becomes a chained NET

$$\frac{x \xrightarrow{3\text{to}2} x'}{\text{net3}(x) \longrightarrow \text{net2}(x')}$$

The last closing rule.
It can be applied in any context.



Summary of reconfiguration rules

A 3hub becomes a 2hub

$$\text{basic3} \xrightarrow{3\text{to}2} \text{basic2}$$

A 3N becomes a 2N (if ...)

$$\frac{x_1 \xrightarrow{3\text{to}2} x'_1 \quad x_2 \xrightarrow{3\text{to}2} x'_2 \quad x_3 \xrightarrow{3\text{to}2} x'_3}{\text{link3}(x_1, x_2, x_3) \xrightarrow{3\text{to}2} \text{link2}(\text{link2}(x'_2, x'_1), x'_3)}$$

A triangular meshed NET becomes a chained NET (if ...)

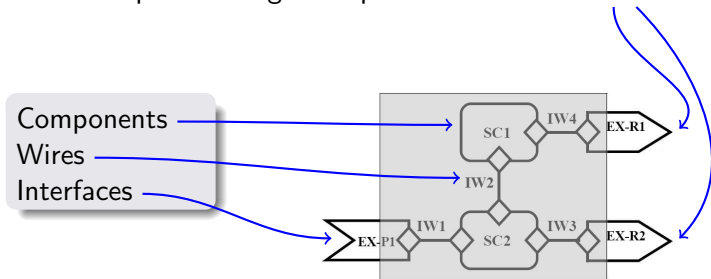
$$\frac{x \xrightarrow{3\text{to}2} x'}{\text{net3}(x) \longrightarrow \text{net2}(x')}$$

Outline

- 1 Introduction
- 2 Architectural Design Rewriting
- 3 Design and Reconfiguration for a Service Modelling Language**
- 4 The end

Design, discover/select, bind...

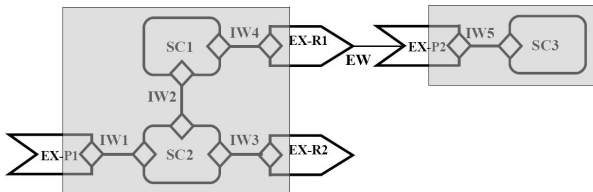
Services are partial designs: required services are holes...



NOTE: Our SML is SRML (Sensoria's SML, inspired by IBM et al.'s Software Component Architecture).

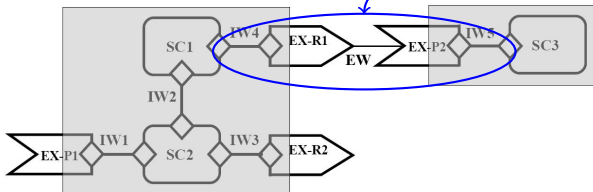
Design, discover/select, bind...

A required service is discovered and selected...



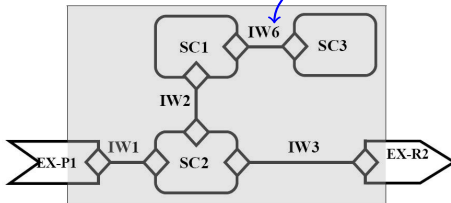
Design, discover/select, **bind**...

Binding reconfigures interfaces and wires...



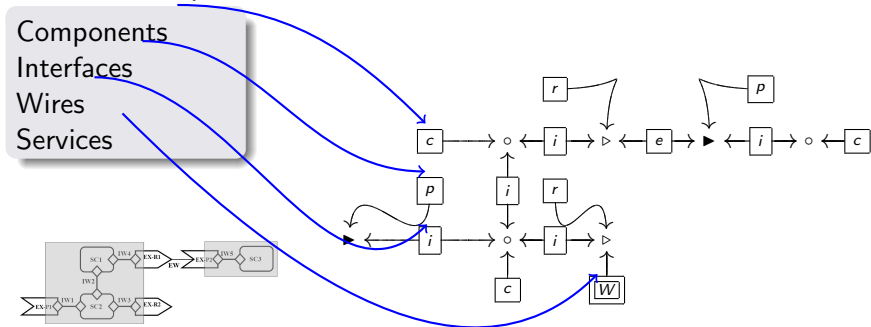
Design, discover/select, **bind**...

The service is bound with a new wire...



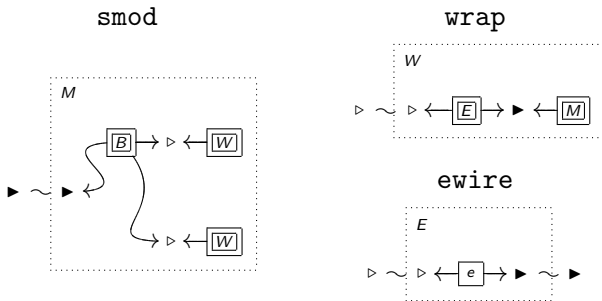
SRML Diagrams as graphs

More or less shaped like this...



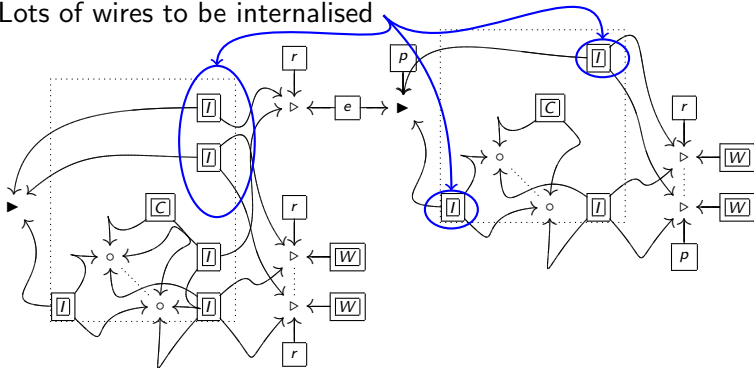
Designing SRML diagrams

An excerpt of the design productions needed



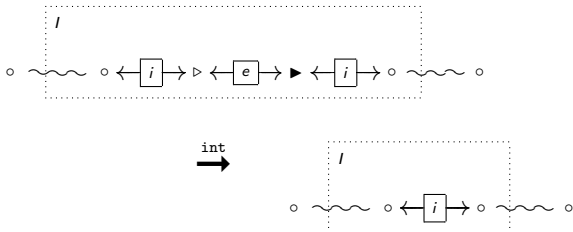
Binding means dealing with things like this...

Lots of wires to be internalised

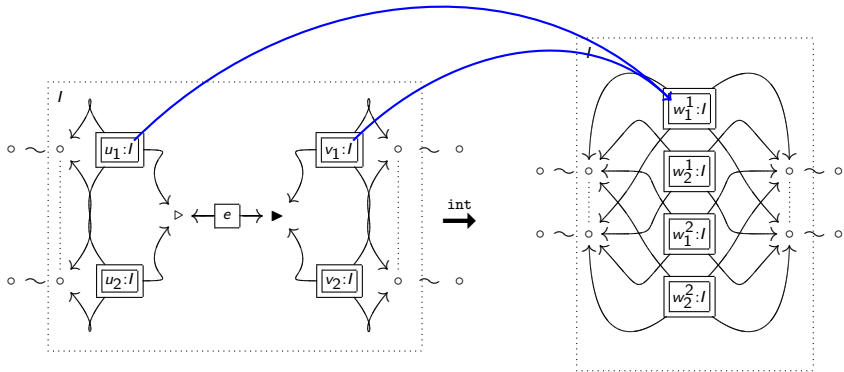


Binding step 1

Reconfiguring a pair of wires...



Binding step 2



Outline

- 1 Introduction
- 2 Architectural Design Rewriting
- 3 Design and Reconfiguration for a Service Modelling Language
- 4 The end**

Concluding remarks

- What is ADR?
 - An algebra of graphs with interfaces (designs).
 - Conditional, labelled rewrite rules on design terms.
- What can I do with ADR?
 - Style-driven design of software architectures.
 - Style-preserving reconfigurations.
 - (Ordinary execution).
 - Architectural Design, Specification and Verification.
 - Run-time Configuration Management.
- Where can I get more on ADR?
 - www.albertolluch.com/adr.html.
 - Style-Based Architectural Reconfigurations, Technical Report TR-07-17, Dipartimento di Informatica, Università di Pisa.