Architectural Design Rewriting as an Architecture Description Language

Ugo Montanari
Università di Pisa

joint work with
Roberto Bruni, Alberto Lluch Lafuente, Univ. Pisa
and Emilio Tuosto, Univ. of Leicester
Rings of Agents with Gateways

The combined load of all the agents of a ring should not exceed a given threshold
Productions
Productions Are Operations

- Design = graph with interface
- Operation chains maps two C-designs to a C-design
  \[ \text{chains}(X:C,Y:C):C \]
- Values of the algebra are software architectures, sorts are nonterminals, carriers are architectural styles
- Ordinary process algebra-like operations on graphs (parallel composition, restriction) are easily represented

\[ \circ x_1 \xrightarrow{w_1} C \xrightarrow{w_2} \circ x_2 \]

\[ \circ x_1 \xrightarrow{\sum} \circ x_3 \xrightarrow{w_3} C \xrightarrow{w_1} \circ x_2 \]
An Algebra of Design Operations

system: $N \Rightarrow S$

net: $R \Rightarrow N$

nets: $N \times N \Rightarrow N$

ring: $C \Rightarrow R$

chains: $C \times C \Rightarrow C$

chain: $\Rightarrow C$
Rings of Agents with Gateways

```
system(nets(net(ring(chains(chain,chain)),net(ring(chain))))): S
```
Partial Designs

Terms with variables represent partial designs
Substitution means refinement; the inverse abstraction
Terms have more information than their evaluations: they are proofs of type

\[
\text{net(}\text{ring(}\text{chains}(X:C,Y:C)))\text{): N} = \text{net(}\text{ring}(Z:C))[\text{chains}(X:C,Y:C):C/Z]\]
Term Rewriting

- Rewriting rules $L \Rightarrow R$ can be instantiated and contextualized.
- Rewritings are concurrent and guaranteed to preserve typing.
- Rewritings can be triggered by the constraint structure:
  e.g. here $w'' \geq c/2$ and $w \geq c/6$ and $w' \geq c/6$, when implied by the present set of constraints, could trigger a duplication of the ring.

$\text{net}(\text{ring}(\text{chains}(X:C,Y:C))):N$

$\Rightarrow$

$\text{nets}(\text{ring}(X:C), \text{ring}(Y:C)):N$
Conditional Rewriting SOS style

- Conditional rules can be of the form:
  \[ X: A \rightarrow a \implies L(X):C \rightarrow b \implies R(Y):D \]

- Complex transitions can be constructed which guarantee synchronous updatings, e.g. nested wrappings for QoS

- Types need not be preserved, but consistent type modifications can be proved

- Process algebra-like semantics with synchronization, extrusion, etc. can be modeled
A Rewriting

\[
\text{ring}(\text{chain}(\text{chain}(C,C),\text{chain}(C,C))):1 \Rightarrow \\
\text{star}(\text{join}(\text{join}(\text{ray}(S),\text{ray}(S)),\text{join}(\text{ray}(S),\text{ray}(S)))):1
\]
Conditional Rewriting Rules

\[ C:2 \Rightarrow \text{ray}(S:2):3 \]

\[ X_1:2 \Rightarrow Y_1:3 \quad X_2:2 \Rightarrow Y_2:3 \]

chain\((X_1, X_2):2 \Rightarrow \text{join}(Y_1, Y_2):3\)

\[ X:2 \Rightarrow Y:3 \]

ring\((X):1 \Rightarrow \text{star}(Y):1\)
Conclusion

- ADR models design, execution and reconfiguration phases
- Process calculi tailored to software architecture
- Presentation in ADR style of:
  - SRML, Sensoria Reference Modeling Language
  - SHR, Synchronized Hyperedge Replacement
  - UML
  - REO, by Farhad Arbab et al., CWI
- SENSORIA case studies about web services
- Implementation in MAUDE
  - Graphical structure
  - Basic operations
  - Reconfiguration as rewriting