# Comparison of Formal Semantics of Reo Connectors: Coloring Models and Constraint Automata

Sung-Shik Jongmans[1] Farhad Arbab[1,2]

[1]Centrum Wiskunde & Informatica (CWI), the Netherlands
[2]Leiden Institute for Advanced Computer Science (LIACS), the Netherlands

9 June 2011

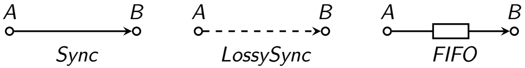
# Table of Contents

## IN A NUTSHELL...

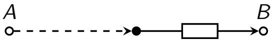
- Introduced in the early 2000s [Arb04].
- Coordinators are called *connectors*.
  - Connectors are directed graphs.
  - Data items flow through edges (or *channels*).
  - Some nodes allow I/O-operations from components:
    - `write(`$d$`,`$n$`)` — send $d$ at $n$
    - $d$`:=take(`$n$`)` — receive $d$ at $n$
- Tool support: modeling, animating, simulating, verifying.

## CONNECTORS & COMPOSITION

- Primitives: connectors without *internal nodes*.
  - Open ended collection.
  - Semantics freely definable:
    - Synchronous or asynchronous;
    - Lossless or lossy;
    - Buffered or non-buffered;
    - ...
  - Examples:




- Composites: connectors obtained through composition.
  - Connectors have compositional semantics.
  - Example:




$\mathit{LossyFIFO} = \mathit{LossySync} \times \mathit{FIFO}$

## Behavioral formalisms

There exist many!

- Coalgebraic models based on streams [AR03].
- Operational models:
  - Constraint automata [BSAR06];
  - Intentional automata [Cos10];
  - Guarded automata [BCS09].
- Coloring models [CCA07]:
  - Two colors;
  - Three colors.
- ...

# Reo

## Behavioral formalisms

There exist many!

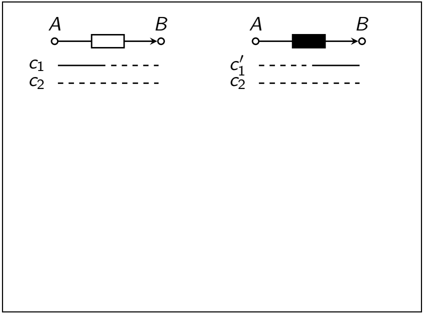
- Coalgebraic models based on streams [AR03].
- Operational models:
  - Constraint automata [BSAR06];
  - Intentional automata [Cos10];
  - Guarded automata [BCS09].
- Coloring models [CCA07]:
  - Two colors;
  - Three colors.
- ...

Folklore

CMs with two colors and CA are equally expressive.

- Yes, but...
- Our contribution:
  - We extend CMs with *data-awareness*.
  - Transformation operator: from data-aware CMs to CA.
  - Transformation operator: from CA to data-aware CMs.
  - Properties / proofs: correctness, compositionality, inverse.

# Coloring Models



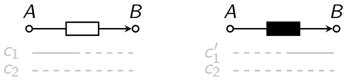

## Colorings

Let $N$ be a set of nodes.

A coloring $c : N \to \mathcal{C}\mathrm{olor}$ over $N$...

- ...maps nodes to colors.
- ...describes a single computation step.

# Coloring Models



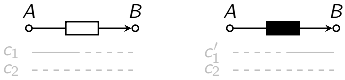

$$S = \left\{ \begin{array}{l} \texttt{FIFO-E} \mapsto \{c_1, c_2\}\ , \\ \texttt{FIFO-F} \mapsto \{c'_1, c_2\} \end{array} \right\}$$

## Coloring table maps

Let $N$ be a set of nodes and $\Lambda$ a set of indexes.
A coloring table map $S : \Lambda \to 2^{\{N \to \mathcal{C}\mathrm{olor}\}}$ over $[N, \Lambda]$...

- ...maps indexes to coloring tables ($=$ sets of colorings).
- ...describes all computation steps in a state.

# Coloring Models




$$S = \begin{Bmatrix} \texttt{FIFO-E} \mapsto \{c_1, c_2\}, \\ \texttt{FIFO-F} \mapsto \{c_1', c_2\} \end{Bmatrix}$$

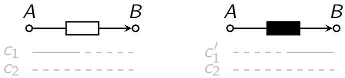
$$\eta = \begin{Bmatrix} \langle \texttt{FIFO-E}, c_1 \rangle \mapsto \texttt{FIFO-F}, \\ \langle \texttt{FIFO-E}, c_2 \rangle \mapsto \texttt{FIFO-E}, \\ \langle \texttt{FIFO-F}, c_1' \rangle \mapsto \texttt{FIFO-E}, \\ \langle \texttt{FIFO-F}, c_2 \rangle \mapsto \texttt{FIFO-F} \end{Bmatrix}$$

## Next functions

Let $S$ be a CTM over $[N, \Lambda]$.
A next function $\eta : \Lambda \times 2^{\{N \to \mathcal{C}\mathrm{olor}\}} \to \Lambda$ over $S$...

- ...maps [indexes, coloring]-pairs to indexes.
- ...describes how the state of a connector evolves over time.

# Coloring Models




$$S = \left\{ \begin{array}{l} \texttt{FIFO-E} \mapsto \{c_1, c_2\}\,, \\ \texttt{FIFO-F} \mapsto \{c_1', c_2\} \end{array} \right\}$$

$$\eta = \left\{ \begin{array}{l} \langle \texttt{FIFO-E}, c_1 \rangle \mapsto \texttt{FIFO-F}\,, \\ \langle \texttt{FIFO-E}, c_2 \rangle \mapsto \texttt{FIFO-E}\,, \\ \langle \texttt{FIFO-F}, c_1' \rangle \mapsto \texttt{FIFO-E}\,, \\ \langle \texttt{FIFO-F}, c_2 \rangle \mapsto \texttt{FIFO-F} \end{array} \right\}$$

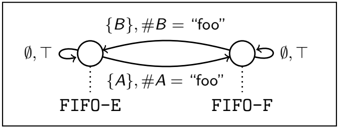
$$\epsilon = \langle \eta, \texttt{FIFO-E} \rangle$$

## Initialized next functions

Let $\eta$ be a next function over $S$ with $S$ defined over $[N, \Lambda]$.
An initialized next function $\epsilon$ over $S$...

- ...associates a next function with an index.
- ...describes the initial state.

# Constraint Automata




## Automata

Let $N$ be a set of nodes and $G$ a set of *data constraints*.

A constraint automaton $\alpha$ over $[N, G]$ is a tuple $\langle Q, R, q_0 \rangle$ with:

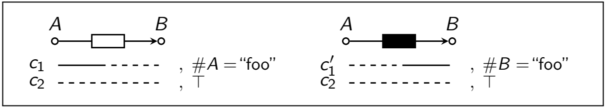
- $Q$ a set of states;
- $R \subseteq Q \times 2^N \times G \times Q$ a transition relation;
- $q_0 \in Q$ an initial state.

# Data-Aware Coloring Models

## Motivation

- CA are data-aware; CMs are not.
- Transformation issue: what to do with constraints?
- One-to-one versus many-to-one.

# Data-Aware Coloring Models




## Extension

- Associate each coloring with a data constraint.
- *Constraint coloring* $\mathbf{c} = \langle c, g \rangle$ over $[N, G]$ with:
  - $c : N \to \mathcal{C}\mathrm{olor}$ a coloring;
  - $g \in G$ a data constraint.
- (Update other constituents of CMs accordingly.)

# From Data-Aware CMs to CA

$$\mathbb{L}(\epsilon) = \langle \Lambda, R, \lambda_0 \rangle$$
with: $R = \{ \langle \lambda, F, g, \boldsymbol{\eta}(\lambda, \langle c, g \rangle) \rangle \mid \lambda \in \Lambda \text{ and } \langle c, g \rangle \in \mathbf{S}(\lambda) \text{ and } F = \{ n \in N \mid c(n) = \text{——} \} \}$

## Definition of $\mathbb{L}$

Let $\epsilon = \langle \boldsymbol{\eta}, \lambda_0 \rangle$ be an initialized (constraint) next function over **S** with **S** defined over $[N, G, \Lambda]$.

- Set of states is the set of indexes $\Lambda$ over which **S** is defined.
- Transition relation includes for each $\langle \lambda, \langle c, g \rangle \rangle \mapsto \lambda' \in \boldsymbol{\eta}$:
  - a transition from $\lambda$ to $\lambda'$,
  - labeled with $g$ as its data constraint and
  - with the set of nodes to which $c$ assigns the flow color.
- Initial state is $\lambda_0$.

# From Data-Aware CMs to CA

$$\mathbb{L}(\epsilon) = \langle \Lambda, R, \lambda_0 \rangle$$
with: $R = \{\, \langle \lambda, F, g, \boldsymbol{\eta}(\lambda, \langle c, g \rangle) \rangle \mid \lambda \in \Lambda$ and $\langle c, g \rangle \in \mathbf{S}(\lambda)$ and $F = \{\, n \in N \mid c(n) = \text{——} \,\} \,\}$

## Definition of $\mathbb{L}$

Let $\boldsymbol{\epsilon} = \langle \boldsymbol{\eta}, \lambda_0 \rangle$ be an initialized (constraint) next function over **S** with **S** defined over $[N, G, \Lambda]$.

- Set of states is the set of indexes $\Lambda$ over which **S** is defined.
- Transition relation includes for each $\langle \lambda, \langle c, g \rangle \rangle \mapsto \lambda' \in \boldsymbol{\eta}$:
  - a transition from $\lambda$ to $\lambda'$,
  - labeled with $g$ as its data constraint and
  - with the set of nodes to which $c$ assigns the flow color.
- Initial state is $\lambda_0$.

# From Data-Aware CMs to CA

$$\mathbb{L}(\epsilon) = \langle \Lambda, R, \lambda_0 \rangle$$
with: $R = \{\, \langle \lambda, F, g, \boldsymbol{\eta}(\lambda, \langle c, g \rangle) \rangle \mid \lambda \in \Lambda$ and $\langle c, g \rangle \in \mathbf{S}(\lambda)$ and $F = \{\, n \in N \mid c(n) = \text{——} \,\} \,\}$

## Definition of $\mathbb{L}$

Let $\boldsymbol{\epsilon} = \langle \boldsymbol{\eta}, \lambda_0 \rangle$ be an initialized (constraint) next function over $\mathbf{S}$ with $\mathbf{S}$ defined over $[N, G, \Lambda]$.

- Set of states is the set of indexes $\Lambda$ over which $\mathbf{S}$ is defined.
- Transition relation includes for each $\langle \lambda, \langle c, g \rangle \rangle \mapsto \lambda' \in \boldsymbol{\eta}$:
  - a transition from $\lambda$ to $\lambda'$,
  - labeled with $g$ as its data constraint and
  - with the set of nodes to which $c$ assigns the flow color.
- Initial state is $\lambda_0$.

# From Data-Aware CMs to CA

$$\mathbb{L}(\epsilon) = \langle \Lambda, R, \lambda_0 \rangle$$
with: $R = \{\, \langle \lambda, F, g, \eta(\lambda, \langle c, g \rangle) \rangle \mid \lambda \in \Lambda \text{ and } \langle c, g \rangle \in \mathbf{S}(\lambda) \text{ and } F = \{\, n \in N \mid c(n) = \text{———} \,\} \,\}$

## Definition of $\mathbb{L}$

Let $\boldsymbol{\epsilon} = \langle \boldsymbol{\eta}, \lambda_0 \rangle$ be an initialized (constraint) next function over $\mathbf{S}$ with $\mathbf{S}$ defined over $[N, G, \Lambda]$.

- Set of states is the set of indexes $\Lambda$ over which $\mathbf{S}$ is defined.
- Transition relation includes for each $\langle \lambda, \langle c, g \rangle \rangle \mapsto \lambda' \in \boldsymbol{\eta}$:
  - a transition from $\lambda$ to $\lambda'$,
  - labeled with $g$ as its data constraint and
  - with the set of nodes to which $c$ assigns the flow color.
- Initial state is $\lambda_0$.

# From Data-Aware CMs to CA

$$\mathbb{L}(\epsilon) = \langle \Lambda, R, \lambda_0 \rangle$$
with: $R = \{ \langle \lambda, F, g, \boldsymbol{\eta}(\lambda, \langle c, g \rangle) \rangle \mid \lambda \in \Lambda \text{ and } \langle c, g \rangle \in \mathbf{S}(\lambda) \text{ and } F = \{ n \in N \mid c(n) = \text{——} \} \}$

## Definition of $\mathbb{L}$

Let $\epsilon = \langle \boldsymbol{\eta}, \lambda_0 \rangle$ be an initialized (constraint) next function over **S** with **S** defined over $[N, G, \Lambda]$.

- Set of states is the set of indexes $\Lambda$ over which **S** is defined.
- Transition relation includes for each $\langle \lambda, \langle c, g \rangle \rangle \mapsto \lambda' \in \boldsymbol{\eta}$:
  - a transition from $\lambda$ to $\lambda'$,
  - labeled with $g$ as its data constraint and
  - with the set of nodes to which $c$ assigns the flow color.
- Initial state is $\lambda_0$.

# From Data-Aware CMs to CA

$$\mathbb{L}(\epsilon) = \langle \Lambda, R, \lambda_0 \rangle$$
with: $R = \{\, \langle \lambda, F, g, \eta(\lambda, \langle c, g \rangle) \rangle \mid \lambda \in \Lambda$ and $\langle c, g \rangle \in \mathbf{S}(\lambda)$ and $F = \{\, n \in N \mid c(n) = \text{——} \,\} \,\}$

## Definition of $\mathbb{L}$

Let $\epsilon = \langle \eta, \lambda_0 \rangle$ be an initialized (constraint) next function over $\mathbf{S}$ with $\mathbf{S}$ defined over $[N, G, \Lambda]$.

- Set of states is the set of indexes $\Lambda$ over which $\mathbf{S}$ is defined.
- Transition relation includes for each $\langle \lambda, \langle c, g \rangle \rangle \mapsto \lambda' \in \eta$:
  - a transition from $\lambda$ to $\lambda'$,
  - labeled with $g$ as its data constraint and
  - with the set of nodes to which $c$ assigns the flow color.
- Initial state is $\lambda_0$.

# From Data-Aware CMs to CA

$$\mathbb{L}(\epsilon) = \langle \Lambda, R, \lambda_0 \rangle$$

with: $R = \{ \langle \lambda, F, g, \boldsymbol{\eta}(\lambda, \langle c, g \rangle) \rangle \mid \lambda \in \Lambda$ and $\langle c, g \rangle \in \mathbf{S}(\lambda)$ and $F = \{ n \in N \mid c(n) = \text{——} \} \}$

## Definition of $\mathbb{L}$

Let $\epsilon = \langle \boldsymbol{\eta}, \lambda_0 \rangle$ be an initialized (constraint) next function over $\mathbf{S}$ with $\mathbf{S}$ defined over $[N, G, \Lambda]$.

- Set of states is the set of indexes $\Lambda$ over which $\mathbf{S}$ is defined.
- Transition relation includes for each $\langle \lambda, \langle c, g \rangle \rangle \mapsto \lambda' \in \boldsymbol{\eta}$:
  - a transition from $\lambda$ to $\lambda'$,
  - labeled with $g$ as its data constraint and
  - with the set of nodes to which $c$ assigns the flow color.
- Initial state is $\lambda_0$.

# From Data-Aware CMs to CA

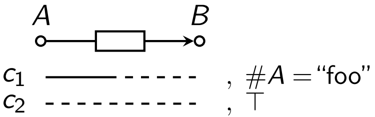
$$\mathbb{L}(\epsilon) = \langle \Lambda, R, \lambda_0 \rangle$$
with: $R = \{ \langle \lambda, F, g, \boldsymbol{\eta}(\lambda, \langle c, g \rangle) \rangle \mid \lambda \in \Lambda \text{ and } \langle c, g \rangle \in \mathbf{S}(\lambda) \text{ and } F = \{ n \in N \mid c(n) = \text{——} \} \}$

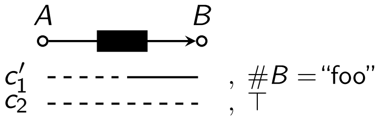
## Definition of $\mathbb{L}$

Let $\boldsymbol{\epsilon} = \langle \boldsymbol{\eta}, \lambda_0 \rangle$ be an initialized (constraint) next function over **S** with **S** defined over $[N, G, \Lambda]$.

- Set of states is the set of indexes $\Lambda$ over which **S** is defined.
- Transition relation includes for each $\langle \lambda, \langle c, g \rangle \rangle \mapsto \lambda' \in \boldsymbol{\eta}$:
  - a transition from $\lambda$ to $\lambda'$,
  - labeled with $g$ as its data constraint and
  - with the set of nodes to which $c$ assigns the flow color.
- Initial state is $\lambda_0$.

# From Data-Aware CMs to CA




$$S = \left\{ \begin{array}{l} \texttt{FIFO-E} \mapsto \{c_1, c_2\}, \\ \texttt{FIFO-F} \mapsto \{c_1', c_2\} \end{array} \right\}$$

$$\eta = \left\{ \begin{array}{l} \langle \texttt{FIFO-E}, c_1 \rangle \mapsto \texttt{FIFO-F}, \\ \langle \texttt{FIFO-E}, c_2 \rangle \mapsto \texttt{FIFO-E}, \\ \langle \texttt{FIFO-F}, c_1' \rangle \mapsto \texttt{FIFO-E}, \\ \langle \texttt{FIFO-F}, c_2 \rangle \mapsto \texttt{FIFO-F} \end{array} \right\}$$

$$\epsilon = \langle \eta, \texttt{FIFO-E} \rangle$$

# From Data-Aware CMs to CA

# From Data-Aware CMs to CA

## Correctness (informal)

We wish to show that a data-aware CM $\epsilon$ and its $\mathbb{L}$-transformation are equivalent: for each mapping in $\epsilon$, there exists a corresponding transition in $\mathbb{L}(\epsilon)$ with respect to...

- the change of state;
- the data items that flow;
- the nodes that fire.

## Compositionality (informal)

We wish to show that it does not matter whether we first compose two coloring models and then apply $\mathbb{L}$ or first apply $\mathbb{L}$ to two coloring models and then compose them.

# From CA to Data-Aware CMs

$$\tfrac{1}{\mathbb{L}}(\alpha) = \langle \boldsymbol{\eta}, q_0 \rangle$$

with: $\boldsymbol{\eta} = \{\, \langle q, \mathbf{c} \rangle \mapsto q' \mid \langle q, F, g, q' \rangle \in R \,\}$
and: $\mathbf{c} = \langle c, g \rangle$
and: $c = \left\{ n \mapsto \kappa \;\middle|\; n \in N \text{ and } \kappa = \left( \begin{array}{ll} \text{———} & \text{if } n \in F \\ \text{- - - -} & \text{otherwise} \end{array} \right) \right\}$

## Definition of $\frac{1}{\mathbb{L}}$

Let $\alpha = \langle Q, R, q_0 \rangle$ be a *deterministic* CA over $[N, G]$.

- Paper: $\frac{1}{\mathbb{L}}$ is correct and compositional.
- Inverse:
    - Lemma: $\frac{1}{\mathbb{L}}(\mathbb{L}(\boldsymbol{\epsilon})) = \boldsymbol{\epsilon}$.
    - Lemma: $\mathbb{L}(\frac{1}{\mathbb{L}}(\alpha)) = \alpha$
    - (The latter cannot hold if we consider data-*un*aware CMs!)

# Concluding Remarks

## Possible applications

- Verification of *context-sensitivity* with Vereofy [BBKK09].
  1. Transform 3CM to 2CM [JKA11].
  2. Transform 2CM to CA with $\mathbb{L}$.
  3. Verify!
- Animation of counterexamples.
  - Vereofy can visualize counterexamples *if* there is a CM.
  - Problem: this is not always the case...
  - Solution: use $\frac{1}{\mathbb{L}}$ to generate unavailable CMs.

# Concluding Remarks

## Summary

- We extended coloring models with data-awareness.
- We defined transformation operators.
- The operators are correct, compositional, and inverse.

## Future work

- Implement operators.
- Implement proposed extensions to Vereofy.
- Investigate other semantic models of Reo.

□

# References I


[AR03] Farhad Arbab and Jan Rutten.
A coinductive calculus of component connectors.
In Marin Wirsing, Dirk Pattinson, and Rolf Hennicker, editors, *Recent Trends in Algebraic Development Techniques*, volume 2755 of *LNCS*, pages 34–55. 2003.

[Arb04] Farhad Arbab.
Reo: A channel-based coordination model for component composition.
*Mathematical Structures in Computer Science*, 14:329–366, 2004.

# References II

[BBKK09] Christel Baier, Tobias Blechmann, Joachim Klein, and Sascha Klüppelholz.
A uniform framework for modeling and verifying components and connectors.
In John Field and Vasco Vasconcelos, editors, *Coordination Models and Languages*, volume 5521 of *Lecture Notes in Computer Science*, pages 247–267. 2009.

[BCS09] Marcello M. Bonsangue, Dave Clarke, and Alexandra Silva.
Automata for context-dependent connectors.
In John Field and Vasco Vasconcelos, editors, *Coordination Models and Languages*, volume 5521 of *LNCS*, pages 184–203. 2009.

[BSAR06] Christel Baier, Marjan Sirjani, Farhad Arbab, and Jan Rutten.
Modeling component connectors in Reo by constraint automata.
*Science of Computer Programming*, 61(2):75–113, 2006.

[CCA07] Dave Clarke, David Costa, and Farhad Arbab.
Connector colouring I: Synchronisation and context dependency.
*Science of Computer Programming*, 66(3):205–225, 2007.

[Cos10] David Costa.
*Formal Models for Component Connectors*.
PhD thesis, Vrije Universiteit Amsterdam, 2010.

# References IV

[JKA11] Sung-Shik Jongmans, Christian Krause, and Farhad Arbab.
Encoding context-sensitivity in Reo into non-context-sensitive semantic models.
In Wolfgang de Meuter and Catalin Roman, editors, *Proceedings of the 13th International Conference on Coordination Models and Languages*, volume 6721 of *LNCS*. Springer, 2011.
To appear.