Team Automata: Overview and Roadmap

Maurice ter Beek

FMT, CNR–ISTI, Pisa, Italy

joint work with **Rolf Hennicker** (LMU, Munich, DE) and **José Proença** (University of Porto, PT)

COORDINATION 2024, Groningen, The Netherlands, June 20th, 2023

Outline

- History of Team Automata
- Team Automata and other Coordination Models
- Recent Results
  - **ICTAC20** Compositionality of Safe Communication in Systems of Team Automata
    Extended Team Automata
  - **FM21** Featured Team Automata
  - **FM23** Can we Communicate? Using Dynamic Logic to Verify Team Automata
    Model Check Team Automata
  - **ICTAC23** Realisability of Global Models of Interaction
    Realisable Team Automata
- Future Work

History of Team Automata

Team Automata

A formalism for interacting component-based systems, whereby multiple sending and receiving actions from concurrent automata can synchronise on certain executions

Team Automata

A formalism for interacting component-based systems, whereby multiple sending and receiving actions from concurrent automata can synchronise on certain executions

First proposed at the 1997 ACM SIGGROUP Conference on Supporting Group Work for modelling components of groupware systems and their interconnections

Formally defined in Computer Supported Cooperative Work (CSCW) — The Journal of Collaborative Computing, in terms of component automata that synchronise

Team Automata

A formalism for interacting component-based systems, whereby multiple sending and receiving actions from concurrent automata can synchronise on certain executions

First proposed at the 1997 ACM SIGGROUP Conference on Supporting Group Work for modelling components of groupware systems and their interconnections

Formally defined in Computer Supported Cooperative Work (CSCW) — The Journal of Collaborative Computing, in terms of component automata that synchronise

Inspired by Input/Output (I/O) automata, inheriting the distinction between internal and external (input and output) actions used for communication with the environment

Technically, an extension of I/O automata, imposing hardly any restrictions on the role of actions in components and composition is not limited to the synchronous product

Selected Publications from 25+ Years

2023	Realisability of Global Models of Interaction	ICTAC'23
2023	Overview on Constrained Multiparty Synchronisation in Team Automata *	FACS'23
2023	Can we Communicate? Using Dynamic Logic to Verify Team Automata	FM'23
2021	Featured Team Automata	FM'21
2020	Compositionality of Safe Communication in Systems of Team Automata	ICTAC'20
2020	Team Automata@Work: On Safe Communication	COORDINATION'20
2017	Communication Requirements for Team Automata	COORDINATION'17
2016	Conditions for Compatibility of Components: The Case of Masters and Slaves	ISoLA'16
2014	On Distributed Cooperation and Synchronised Collaboration	JALC
2013	Compatibility in a multi-component environment *	TCS
2012	Vector Team Automata	TCS
2010	Team Automata Based Framework for Spatio-Temporal RBAC Model *	BAIP'10
2009	Associativity of Infinite Synchronized Shuffles and Team Automata	Fundam. Inform.
2008	Extending Team Automata to Evaluate Software Architectural Design *	COMPSAC'08
2008	A calculus for team automata	ENTCS
2007	A Review on Specifying Software Architectures Using Extended Automata-Based Models *	FSEN'07
2006	Modelling a Secure Agent with Team Automata *	ENTCS
2006	A Team Automaton Scenario for the Analysis of Security Properties in Communication Protocols	JALC
2005	Team Automata for Security – A Survey –	ENTCS
2005	Modularity for Teams of I/O Automata	IPL
2004	Teams of Pushdown Automata	IJCM
2004	Interactive Behaviour of Multi-Component Systems *	ToBaCo'04
2003	Team Automata: A Formal Approach to the Modeling of Collaboration Between System Components	PhD thesis
2003	Team Automata Satisfying Compositionality	FME'03
2003	Team Automata for CSCW – A Survey – *	LNCS
2003	Synchronizations in Team Automata for Groupware Systems	CSCW
2002	Towards Team-Automata-Driven Object-Oriented Collaborative Work *	LNCS
2001	Team Automata for Spatial Access Control	ECSCW'01
1997	Team Automata for Groupware Systems	GROUP'97

2023	Realisability of Global Models of Interaction	ICTAC'23
2023	Overview on Constrained Multiparty Synchronisation in Team Automata *	FACS'23
2023	Can we Communicate? Using Dynamic Logic to Verify Team Automata	FM'23
2021	Featured Team Automata	FM'21
2020	Compositionality of Safe Communication in Systems of Team Automata	ICTAC'20
2020	Team Automata@Work: On Safe Communication	COORDINATION'20
2017	Communication Requirements for Team Automata	COORDINATION'17
2016	Conditions for Compatibility of Components: The Case of Masters and Slaves	ISoLA'16
2014	On Distributed Cooperation and Synchronised Collaboration	JALC
2013	Compatibility in a multi-component environment *	TCS
2012	Vector Team Automata	TCS
2010	Team Automata Based Framework for Spatio-Temporal RBAC Model *	BAIP'10
2009	Associativity of Infinite Synchronized Shuffles and Team Automata	Fundam. Inform.
2008	Extending Team Automata to Evaluate Software Architectural Design *	COMPSAC'08
2008	A calculus for team automata	ENTCS
2007	A Review on Specifying Software Architectures Using Extended Automata-Based Models *	FSEN'07
2006	Modelling a Secure Agent with Team Automata *	ENTCS
2006	A Team Automaton Scenario for the Analysis of Security Properties in Communication Protocols	JALC
2005	Team Automata for Security – A Survey –	ENTCS
2005	Modularity for Teams of I/O Automata	IPL
2004	Teams of Pushdown Automata	IJCM
2004	Interactive Behaviour of Multi-Component Systems *	ToBaCo'04
2003	Team Automata: A Formal Approach to the Modeling of Collaboration Between System Components	PhD thesis
2003	Team Automata Satisfying Compositionality	FME'03
2003	Team Automata for CSCW – A Survey – *	LNCS
2003	Synchronizations in Team Automata for Groupware Systems	CSCW
2002	Towards Team-Automata-Driven Object-Oriented Collaborative Work *	LNCS
2001	Team Automata for Spatial Access Control	ECSCW'01
1997	Team Automata for Groupware Systems *	GROUP'97

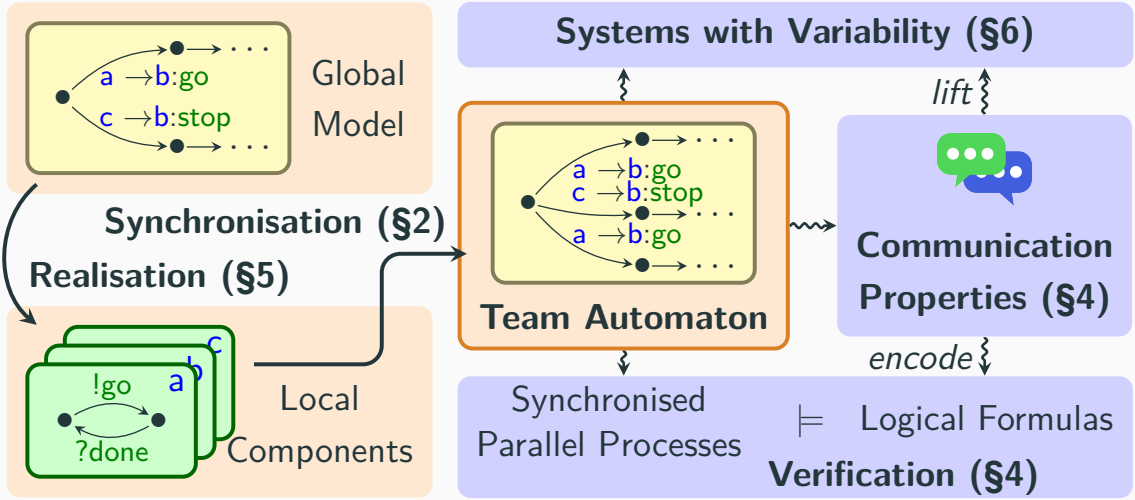
2023	Realisability of Global Models of Interaction	ICTAC'23
2023	Overview on Constrained Multiparty Synchronisation in Team Automata *	FACS'23
2023	Can we Communicate? Using Dynamic Logic to Verify Team Automata	FM'23
2021	Featured Team Automata	FM'21
2020	Compositionality of Safe Communication in Systems of Team Automata	ICTAC'20
2020	Team Automata@Work: On Safe Communication	COORDINATION'20
2017	Communication Requirements for Team Automata	COORDINATION'17
2016	Conditions for Compatibility of Components: The Case of Masters and Slaves	ISoLA'16
2014	On Distributed Cooperation and Synchronised Collaboration	JALC
2013	Compatibility in a multi-component environment *	TCS
2012	Vector Team Automata	TCS
2010	Team Automata Based Framework for Spatio-Temporal RBAC Model *	BAIP'10
2009	Associativity of Infinite Synchronized Shuffles and Team Automata	Fundam. Inform.
2008	Extending Team Automata to Evaluate Software Architectural Design *	COMPSAC'08
2008	A calculus for team automata	ENTCS
2007	A Review on Specifying Software Architectures Using Extended Automata-Based Models *	FSEN'07
2006	Modelling a Secure Agent with Team Automata *	ENTCS
2006	A Team Automaton Scenario for the Analysis of Security Properties in Communication Protocols	JALC
2005	Team Automata for Security – A Survey –	ENTCS
2005	Modularity for Teams of I/O Automata	IPL
2004	Teams of Pushdown Automata	IJCM
2004	Interactive Behaviour of Multi-Component Systems *	ToBaCo'04
2003	Team Automata: A Formal Approach to the Modeling of Collaboration Between System Components	PhD thesis
2003	Team Automata Satisfying Compositionality	FME'03
2003	Team Automata for CSCW – A Survey – *	LNCS
2003	Synchronizations in Team Automata for Groupware Systems	CSCW
2002	Towards Team-Automata-Driven Object-Oriented Collaborative Work *	LNCS
2001	Team Automata for Spatial Access Control	ECSCW'01
1997	Team Automata for Groupware Systems *	GROUP'97

2023	Realisability of Global Models of Interaction	ICTAC'23
2023	Overview on Constrained Multiparty Synchronisation in Team Automata	FACS'23
2023	Can we Communicate? Using Dynamic Logic to Verify Team Automata	FM'23
2021	Featured Team Automata	FM'21
2020	Compositionality of Safe Communication in Systems of Team Automata	ICTAC'20
2020	Team Automata@Work: On Safe Communication	COORDINATION'20
2017	Communication Requirements for Team Automata	COORDINATION'17
2016	Conditions for Compatibility of Components: The Case of Masters and Slaves	ISoLA'16
2014	On Distributed Cooperation and Synchronised Collaboration	JALC
2013	Compatibility in a multi-component environment *	TCS
2012	Vector Team Automata	TCS
2010	Team Automata Based Framework for Spatio-Temporal RBAC Model *	BAIP'10
2009	Associativity of Infinite Synchronized Shuffles and Team Automata	Fundam. Inform.
2008	Extending Team Automata to Evaluate Software Architectural Design *	COMPSAC'08
2008	A calculus for team automata	ENTCS
2007	A Review on Specifying Software Architectures Using Extended Automata-Based Models *	FSEN'07
2006	Modelling a Secure Agent with Team Automata *	ENTCS
2006	A Team Automaton Scenario for the Analysis of Security Properties in Communication Protocols	JALC
2005	Team Automata for Security – A Survey –	ENTCS
2005	Modularity for Teams of I/O Automata	IPL
2004	Teams of Pushdown Automata	IJCM
2004	Interactive Behaviour of Multi-Component Systems *	ToBaCo'04
2003	Team Automata: A Formal Approach to the Modeling of Collaboration Between System Components	PhD thesis
2003	Team Automata Satisfying Compositionality	FME'03
2003	Team Automata for CSCW – A Survey – *	LNCS
2003	Synchronizations in Team Automata for Groupware Systems	CSCW
2002	Towards Team-Automata-Driven Object-Oriented Collaborative Work *	LNCS
2001	Team Automata for Spatial Access Control	ECSCW'01
1997	Team Automata for Groupware Systems *	GROUP'97

Coordination Formalisms and Aspects Analysed in the Paper

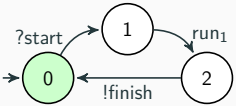
Coordination Formalism (Sections 2 & 3)	Communication Properties (Section 4)	Realisability (Section 5)	Verification (Sections 4 & 4)	Supporting Tools (Sections 3, 4 & 5)	Variability (Section 6)	Data (Section 6)
Team Automata	✓	✓	✓	✓	✓	
Reo via Port Automata			✓	✓	✓	✓
BIP			✓	✓	✓	✓
Contract Automata	✓	✓	✓	✓	✓	
Choreography Automata	✓	✓	✓	✓		
Multi-Party Session Types		✓	✓		✓	✓

Team Automata Aspects Addressed in the Paper

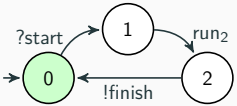


Team Automata by Example

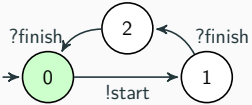
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



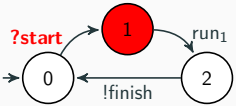
Controller

Team Automata

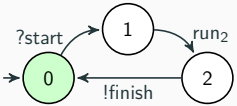
[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

Team Automata by Example

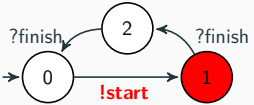
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



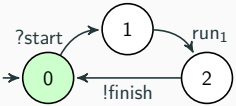
Controller

Team Automata

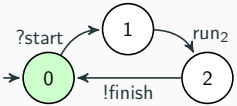
[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

Team Automata by Example

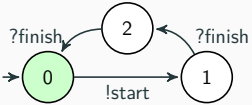
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



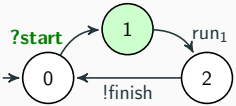
Controller

Team Automata

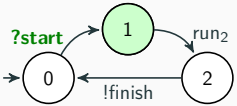
[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

Team Automata by Example

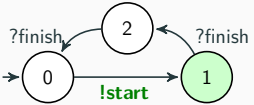
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



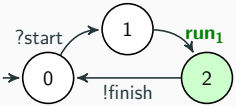
Controller

Team Automata

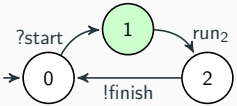
[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

Team Automata by Example

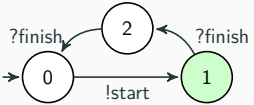
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



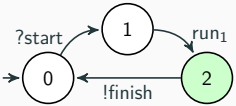
Controller

Team Automata

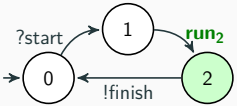
[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

Team Automata by Example

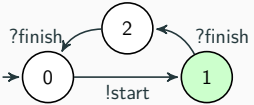
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



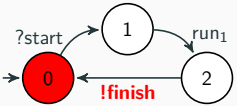
Controller

Team Automata

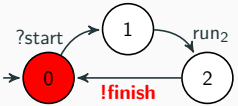
[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

Team Automata by Example

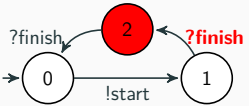
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

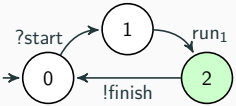


Controller

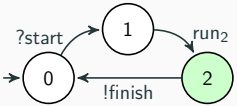
Team Automata

[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

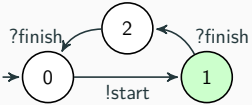
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



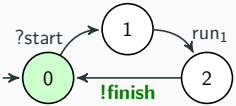
Controller

Team Automata

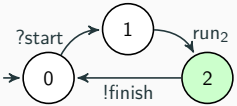
[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

Team Automata by Example

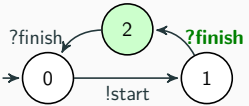
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



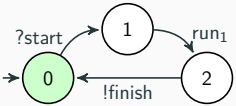
Controller

Team Automata

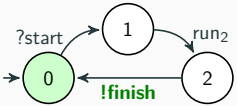
[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

Team Automata by Example

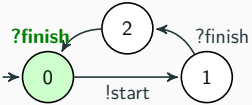
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

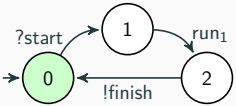


Controller

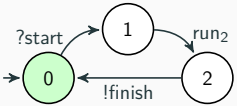
Team Automata

[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

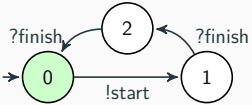
Extended Team Automata: Constrained Multiparty Synchronisations



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

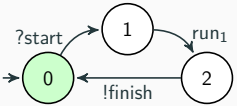


Controller

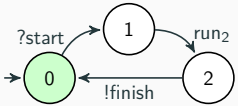
Extended TA synchronisations:

[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

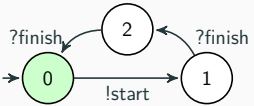
Extended Team Automata: Constrained Multiparty Synchronisations



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



Controller

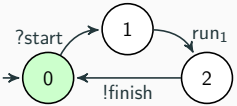
Extended TA synchronisations:

[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

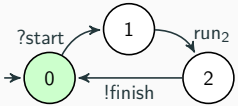
multiparty

Ctr $\rightarrow \{R1, R2\}$: start

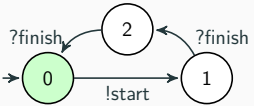
Extended Team Automata: Constrained Multiparty Synchronisations



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



Controller

Extended TA synchronisations:

[CSCW'01'03] [FM'03,'21,'23] [TCS'12'13]
[COORDINATION'17,'20] [ICTAC'20,'23]

multiparty

Ctr $\rightarrow \{R1, R2\}$: start

constrained

start: $1 \rightarrow 2$
finish: $1 \rightarrow 1$

Extended Team Automata and other Coordination Models

Overview on Constrained Multiparty Synchronisation in Team Automata

and other coordination models:

J. Proença, Overview on Constrained Multiparty Synchronisation in Team Automata @ FACS'23

Overview on Constrained Multiparty Synchronisation in Team Automata

and other coordination models:

J. Proença, Overview on Constrained Multiparty Synchronisation in Team Automata @ FACS'23

Runners with orchestrators

- Reo
- BIP

↳ S.-S.T.Q. Jongmans and F. Arbab, Overview of thirty semantic formalisms for Reo. *Scientific Annals of Computer Science* 22 (2012)

S. Bliudze and J. Sifakis, The algebra of connectors: structuring interaction in BIP. *IEEE Transactions on Computers* 57 (2008)

Overview on Constrained Multiparty Synchronisation in Team Automata

and other coordination models:

J. Proença, Overview on Constrained Multiparty Synchronisation in Team Automata @ FACS'23

Runners with orchestrators

- Reo
- BIP

↳ S.-S.T.Q. Jongmans and F. Arbab, Overview of thirty semantic formalisms for Reo. *Scientific Annals of Computer Science* 22 (2012)

S. Bliudze and J. Sifakis, The algebra of connectors: structuring interaction in BIP. *IEEE Transactions on Computers* 57 (2008)

Runners with choreographies

- Choreography Automata
- Multiparty Session Types

↳ F. Barbanera, I. Lanese, and E. Tuosto, Choreography Automata @ COORDINATION'20

S. Ghilezan, S. Jakšić, J. Pantović, A. Scalas, and N. Yoshida, Precise subtyping for synchronous multiparty sessions. *JLAMP* 104 (2019)

Overview on Constrained Multiparty Synchronisation in Team Automata

and other coordination models:

J. Proença, Overview on Constrained Multiparty Synchronisation in Team Automata @ FACS'23

Runners with orchestrators

- Reo
- BIP

Runners with choreographies

- Choreography Automata
- Multiparty Session Types

. . . with both

- Contract Automata

S.-S.T.Q. Jongmans and F. Arbab, Overview of thirty semantic formalisms for Reo. *Scientific Annals of Computer Science* 22 (2012)

S. Bliudze and J. Sifakis, The algebra of connectors: structuring interaction in BIP. *IEEE Transactions on Computers* 57 (2008)

D. Basile, P. Degano, G. Ferrari, and E. Tuosto, Relating two automata-based models of orchestration and choreography. *JLAMP* 85 (2016)

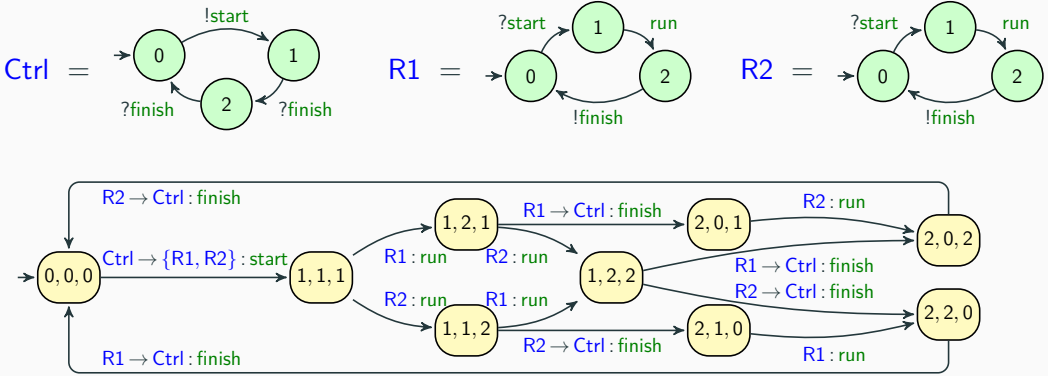
F. Barbanera, I. Lanese, and E. Tuosto, Choreography Automata @ COORDINATION'20

S. Ghilezan, S. Jakšić, J. Pantović, A. Scalas, and N. Yoshida, Precise subtyping for synchronous multiparty sessions. *JLAMP* 104 (2019)

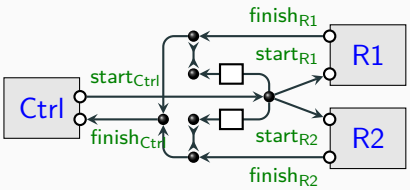
In the paper we discuss for each formalism (1) the definition, means of (2) composition (via synchronisation), (3) a model of the Race example, (4) a brief relation with team automata, and (5) tool support

Team Automata

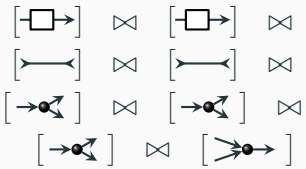
In the paper we discuss for each formalism (1) the definition, means of (2) composition (via synchronisation), (3) a model of the Race example, (4) a brief relation with team automata, and (5) tool support



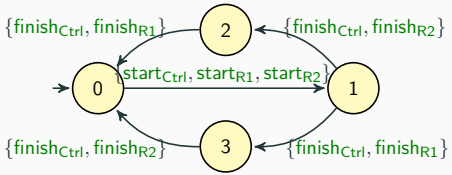
Reo via Port Automata



- Focus on connectors (not on components)
- Connectors built compositionally
- Components should be flexible/compatible

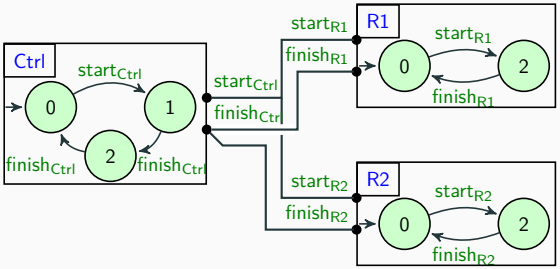


=



(semantics as a port automaton, after hiding internal ports shared among sub-connectors)

BIP without Priorities



$$\mathbb{N} = \{\mathsf{Ctrl}, \mathsf{R1}, \mathsf{R2}\}$$

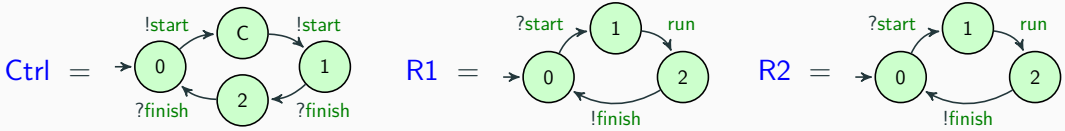
$$BP = (\{B_{\mathsf{Ctrl}}, B_{\mathsf{R1}}, B_{\mathsf{R2}}\}\ ,\ I)$$

$$I = \left\{ \begin{array}{l} \{\mathsf{start}_{\mathsf{Ctrl}}, \mathsf{start}_{\mathsf{R1}}, \mathsf{start}_{\mathsf{R2}}\}, \\ \{\mathsf{finish}_{\mathsf{Ctrl}}, \mathsf{finish}_{\mathsf{R1}}\}, \\ \{\mathsf{finish}_{\mathsf{Ctrl}}, \mathsf{finish}_{\mathsf{R2}}\} \end{array} \right\}$$

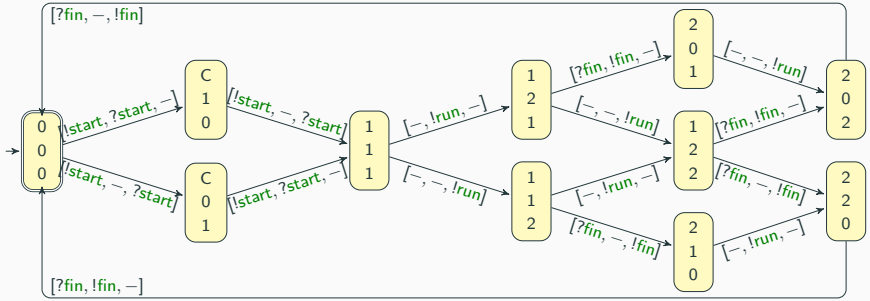
- Components expose ports
- Interactions restrict which ports can fire
- Constructors using unicast (●) and broadcast (▲) can be used to restrict interactions
- Dataflow can be added

(individual components labelled by actions, restricted to the interactions allowed by I)

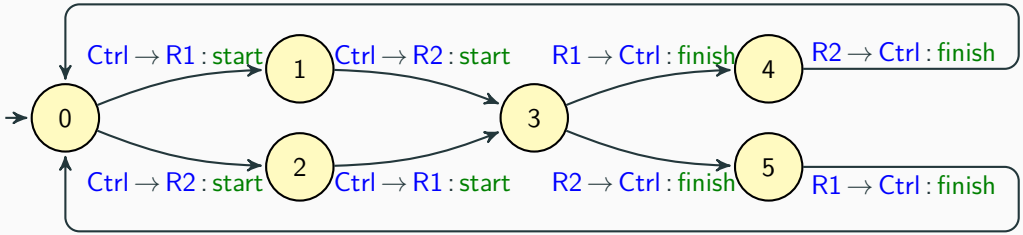
Contract Automata



(**C**ommitted states: whenever the intermediate state of two concatenated transitions is committed, the two transitions are executed atomically)



Choreography Automata



(without internal run actions)

- Similar to contract automata
- Several results over the language of CA
- Realising = Projecting the language of CA

$$\mathcal{G} = \mathsf{Ctrl} \to \mathsf{R1} : \mathsf{start}.\mathsf{Ctrl} \to \mathsf{R2} : \left\{ \begin{array}{l} \mathsf{start}_1.(\mathsf{R1} \to \mathsf{Ctrl} : \mathsf{finish}.\mathsf{R2} \to \mathsf{Ctrl} : \mathsf{finish}.\mathcal{G}), \\ \mathsf{start}_2.(\mathsf{R2} \to \mathsf{Ctrl} : \mathsf{finish}.\mathsf{R1} \to \mathsf{Ctrl} : \mathsf{finish}.\mathcal{G}) \end{array} \right\}$$

$$\mathcal{S} = \mathsf{Ctrl} \rhd \mathcal{L}_{\mathsf{Ctrl}} \parallel \mathsf{R1} \rhd \mathcal{L}_{\mathsf{R1}} \parallel \mathsf{R2} \rhd \mathcal{L}_{\mathsf{R2}}$$

$$\mathcal{L}_{\mathsf{Ctrl}} = \mathsf{R1}!\mathsf{start}.\mathsf{R2}!\left\{\mathsf{start}_1.(\mathsf{R1}?\mathsf{finish}.\mathsf{R2}?\mathsf{finish}.\mathcal{L}_{\mathsf{Ctrl}})\,,\ \mathsf{start}_2.(\mathsf{R2}?\mathsf{finish}.\mathsf{R1}?\mathsf{finish}.\mathcal{L}_{\mathsf{Ctrl}})\right\}$$

$$\mathcal{L}_{\mathsf{R1}} = \mathsf{Ctrl}?\mathsf{start}.\mathsf{Ctrl}!\mathsf{finish}.\mathcal{L}_{\mathsf{R1}}$$

$$\mathcal{L}_{\mathsf{R2}} = \mathsf{Ctrl}?\left\{\mathsf{start}_1.\mathsf{Ctrl}!\mathsf{finish}.\mathcal{L}_{\mathsf{R2}}\,,\ \mathsf{start}_2.\mathsf{Ctrl}!\mathsf{finish}.\mathcal{L}_{\mathsf{R2}}\right\}$$

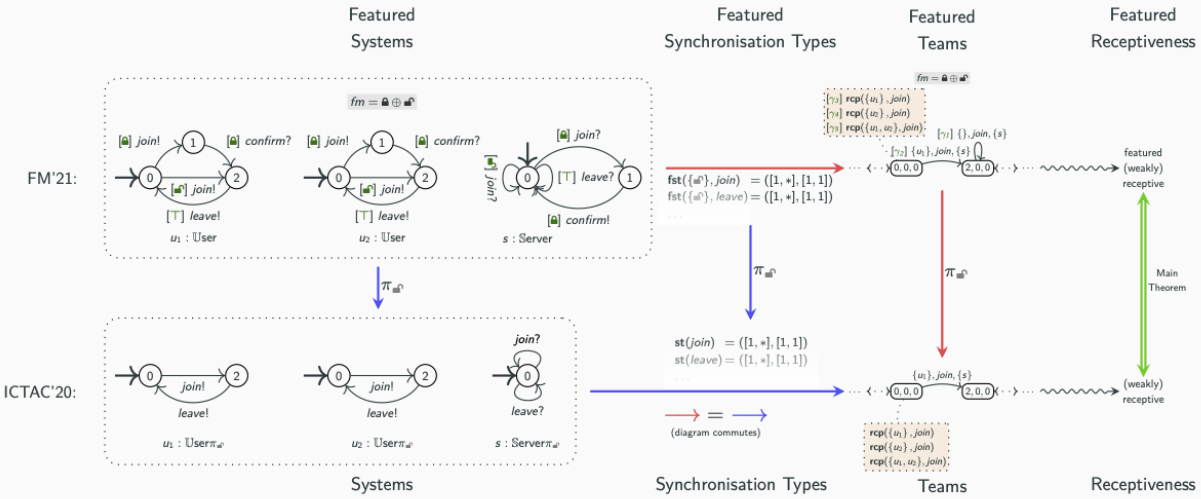
(using only binary synchronisation and
using distinct start_1 and start_2 to differentiate the choice in the branch)

- Use projections for realisation
- Often impose syntactic restrictions on global types

Recent Results

Featured Team Automata

Featured Extended Team Automata (fETA)

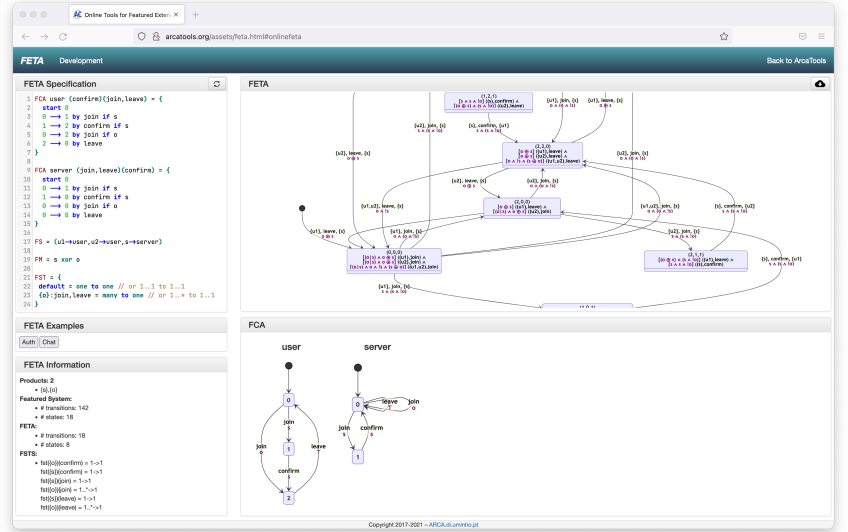


Online prototype: `http://arcatools.org/feta`

Online Prototype: `http://arcatools.org/feta`

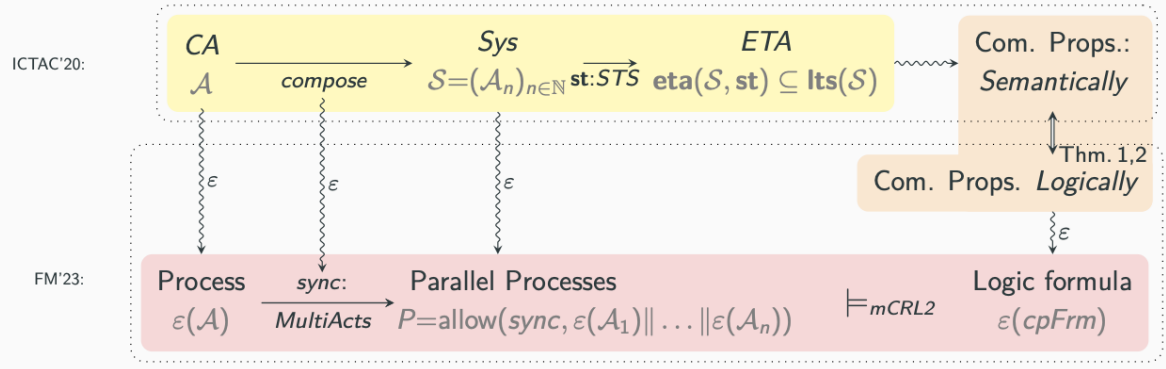
- Specify
- Generate*
- Visualise
- Statistics

*SAT solver to solve *fm*



Model Check Team Automata

Safe communication: receptiveness (*no message loss*) & responsiveness (*no indefinite waiting*)



Online Prototype: https://github.com/arcalab/team-a

ETA Specification

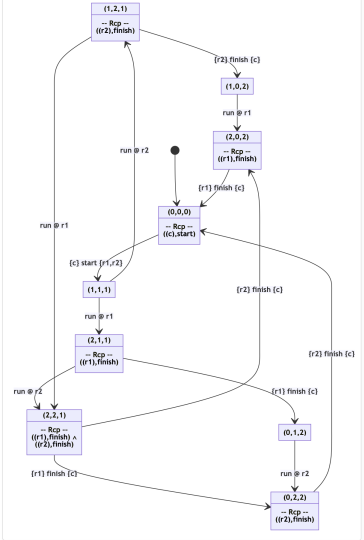
```
1  //Race example
2  CA runner (start)
3                (finish) = {
4    start 0
5    0 --> 1 by start
6    1 --> 2 by run
7    2 --> 0 by finish
8  }
9  CA controller (finish)
10                 (start) = {
11   start 0
12   0 --> 1 by start
13   1 --> 2 by finish
14   2 --> 0 by finish
15 }
16 S = (r1:runner, r2:runner,
17      c:controller)
18 STS = {
19   default = 1 to 1
20   start = 1 to 2
21 }
```

Race example

ETA Examples

Simple | Race | Chat

ETA diagram



Online Prototype: https://github.com/arcalab/team-a

ETA Specification

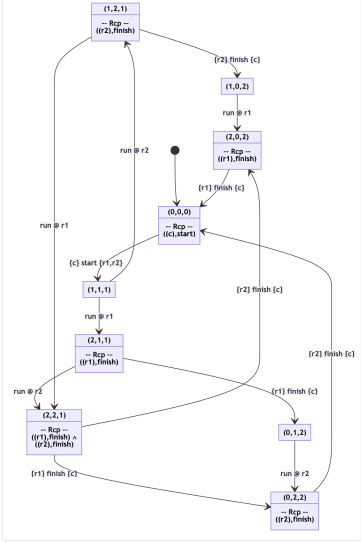
```
1  //Race example
2  CA runner (start)
3           (finish) = {
4    start 0
5    0 --> 1 by start
6    1 --> 2 by run
7    2 --> 0 by finish
8  }
9  CA controller (finish)
10                (start) = {
11   start 0
12   0 --> 1 by start
13   1 --> 2 by finish
14   2 --> 0 by finish
15 }
16 S = (r1:runner, r2:runner,
17       c:controller)
18 STS = {
19  default = 1 to 1
20  start = 1 to 2
21 }
```

Race example

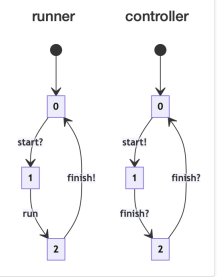
ETA Examples

Simple | Race | Chat

ETA diagram



CA



Online Prototype: `https://github.com/arcalab/team-a`

Communication Properties' Characterisation in mCRL2

Receptiveness:

```
[ (r1_finish|c_finish + r2_run + c_start|r1_start|r2_start + r2_finish|c_finish + r1_run)* ](
  ((<c_start> true)  =>  (<c_start|r1_start|r2_start> true)) &&
  ((<r1_finish> true)  =>  (<r1_finish|c_finish> true)) &&
  ((<r2_finish> true)  =>  (<r2_finish|c_finish> true))
)
```

Responsiveness:

```
[ (r1_finish|c_finish + r2_run + c_start|r1_start|r2_start + r2_finish|c_finish + r1_run)* ](
   (<c_finish +
     r1_start|r2_start> true)
   =>
   (<r1_finish|c_finish +
     c_start|r1_start|r2_start +
     r2_finish|c_finish> true)
)
```

Weak Receptiveness:

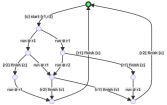
```
[ (r1_finish|c_finish + r2_run + c_start|r1_start|r2_start + r2_finish|c_finish + r1_run)* ](
  ((<r1_finish> true)  =>  (<(r2_run+r2_finish|c_finish)* . r1_finish|c_finish> true)) &&
  ((<r2_finish> true)  =>  (<(r1_finish|c_finish+r1_run)* . r2_finish|c_finish> true)) &&
  ((<c_start> true)  =>  (<(r2_run+r1_run)* . c_start|r1_start|r2_start> true))
)
```

Weak Responsiveness:

```
[ (r1_finish|c_finish + r2_run + c_start|r1_start|r2_start + r2_finish|c_finish + r1_run)* ](
   (<c_finish +
     r1_start|r2_start> true)
   =>
   (<(r2_run+r1_run)* . r1_finish|c_finish +
     c_start|r1_start|r2_start +
     (r2_run+r1_run)* . r2_finish|c_finish> true)
)
```

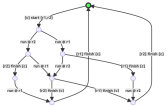
View mCRL2 evidence

Receptiveness: true

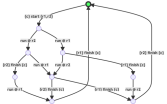


Responsiveness: false

Weak Receptiveness: true



Weak Responsiveness: true



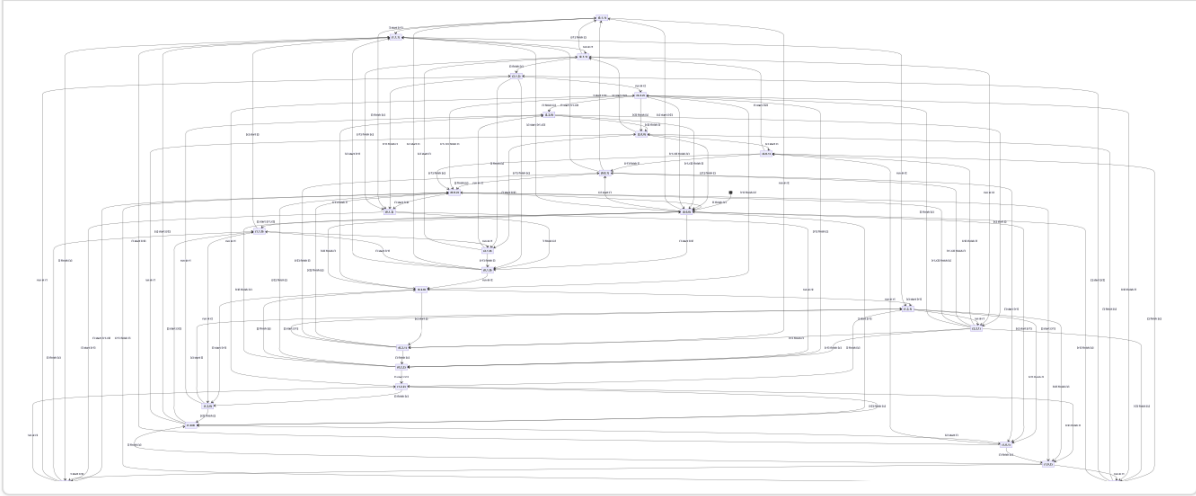
Online Prototype: https://github.com/arcalab/team-a

mCRL2 full system:

```
act
   r1_start,c_finish,r2_run,c_start,r2_finish,r2_start,r1_finish,r1_run;
proc
   r1(s:Int) =
     (s == 2) -> ( r1_finish.r1(0) ) +
     (s == 1) -> ( r1_run.r1(2) ) +
     (s == 0) -> ( r1_start.r1(1) );
   r2(s:Int) =
     (s == 2) -> ( r2_finish.r2(0) ) +
     (s == 1) -> ( r2_run.r2(2) ) +
     (s == 0) -> ( r2_start.r2(1) );
   c(s:Int) =
     (s == 2) -> ( c_finish.c(0) ) +
     (s == 1) -> ( c_finish.c(2) ) +
     (s == 0) -> ( c_start.c(1) );
init
   allow({
     r1_start,
     c_finish,
     c_start|r2_start,
     c_start,
     c_start|r2_start|r1_start,
     r2_finish|r1_finish,
     c_finish|r2_finish,
     r2_finish,
     r2_start,
     r1_finish,
     c_start|r1_start,
     r1_run,
     c_finish|r2_finish|r1_finish,
     r2_run,
     r2_start|r1_start,
     c_finish|r1_finish},
   r1(0) || r2(0) || c(0));
```

Online Prototype: https://github.com/arcalab/team-a

System diagram



Realisable Team Automata

Challenge

How to check if a global model is realisable and, if it is, how to synthesise a realisation?

Challenge

How to check if a global model is realisable and, if it is, how to synthesise a realisation?

$$\mathcal{M} \textit{ realised by } \mathcal{S} = (\mathcal{M}_i)_{i \in \mathcal{I}} ?$$

Solutions typically impose syntactic restrictions on global types, using projections to obtain local models:

F. Barbanera, I. Lanese, and E. Tuosto, Formal Choreographic Languages @ COORDINATION'22

M. Hüttel et al., Foundations of Session Types and Behavioural Contracts. *ACM Comput. Surv.* 49 (2016)

Challenge

How to check if a global model is realisable and, if it is, how to synthesise a realisation?

$$\mathcal{M} \textit{ realised by } \mathcal{S} = (\mathcal{M}_i)_{i \in \mathcal{I}} \, ?$$

Solutions typically impose syntactic restrictions on global types, using projections to obtain local models:

F. Barbanera, I. Lanese, and E. Tuosto, Formal Choreographic Languages @ COORDINATION'22

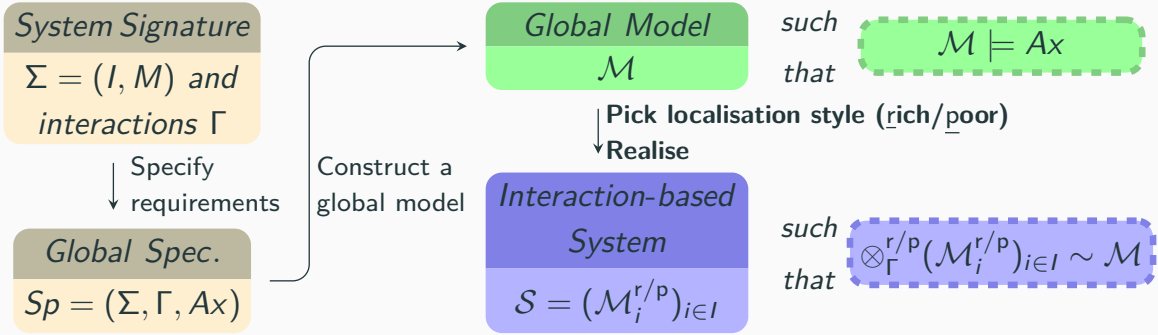
M. Hüttel et al., Foundations of Session Types and Behavioural Contracts. *ACM Comput. Surv.* 49 (2016)

$$\otimes(\mathcal{M}_i)_{i \in \mathcal{I}} \models Sp \, ?$$

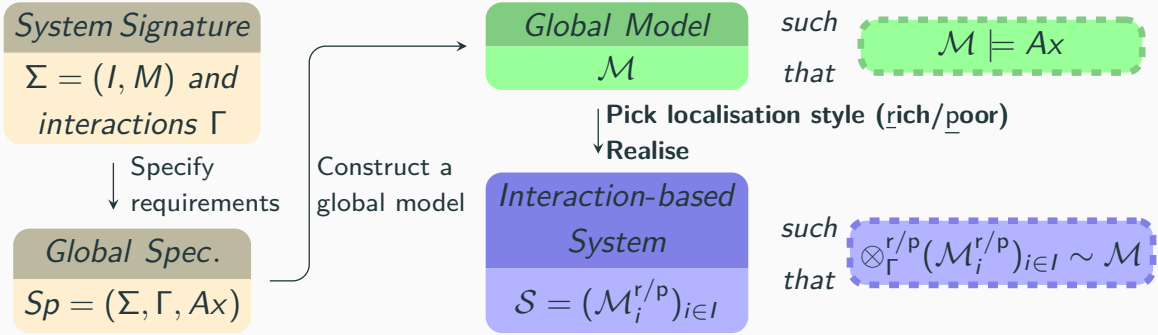
Alternatively, provide a specification in some logical formalism, and construct local models from scratch:

R. Hennicker, Role-Based Development of Dynamically Evolving Esembles @ WADT'18

Contributions ICTAC'23



Contributions ICTAC'23



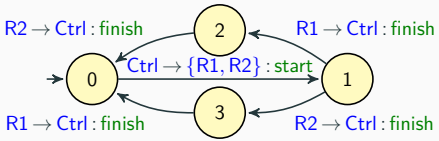
Multi-interactions

rich (à la multi-party session types, choreography languages) $i \to j : m \Rightarrow$ local output action $ij!m$ for i and local input action $ij?m$ for j

poor (à la component-based I/O development, loose coupling) $i \to j : m \Rightarrow$ local output action $!m$ for i and local input action $?m$ for j

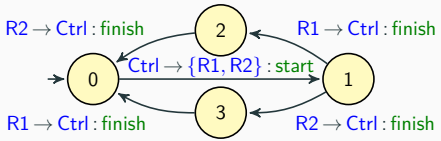
Example

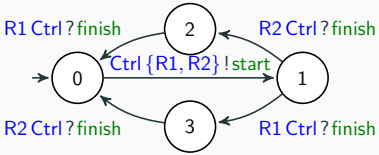
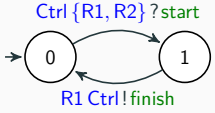
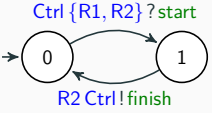
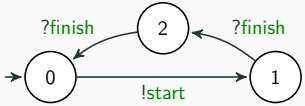
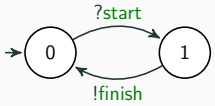
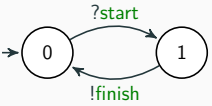
$$\Gamma_{\text{Race}} = \left\{ \begin{array}{l} \text{Ctrl} \to \{\text{R1}, \text{R2}\} : \text{start}\,, \\ \text{R1} \to \text{Ctrl} : \text{finish}\,, \\ \text{R2} \to \text{Ctrl} : \text{finish} \end{array} \right\}$$



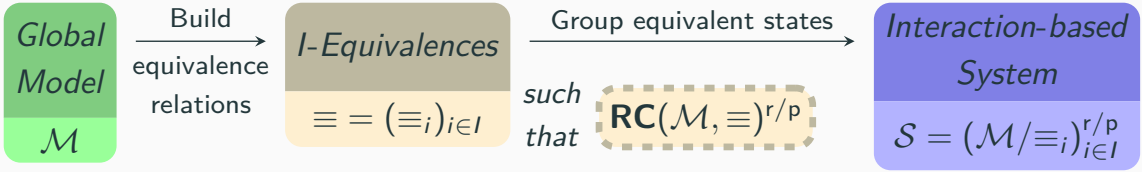
Example

$$\Gamma_{\text{Race}} = \left\{ \begin{array}{l} \text{Ctrl} \to \{\text{R1}, \text{R2}\} : \text{start}\,, \\ \text{R1} \to \text{Ctrl} : \text{finish}\,, \\ \text{R2} \to \text{Ctrl} : \text{finish} \end{array} \right\}$$

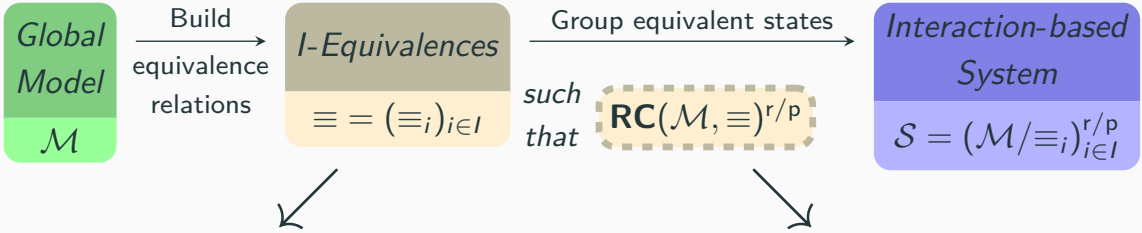


Localisation	Local Ctrl	Local R1	Local R2
Rich			
Poor			

Check Realisability and System Synthesis



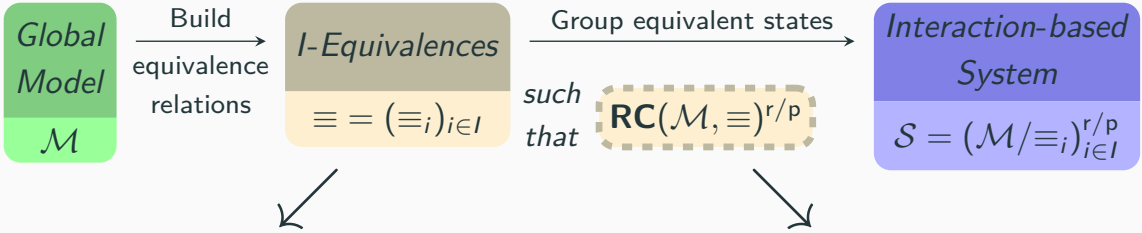
Check Realisability and System Synthesis



I. Castellani, M. Mukund, and P.S. Thiagarajan,
Synthesizing Distributed Transition Systems
from Global Specifications @ FSTTCS'99

cf. our paper for details:
M.H. ter Beek, R. Hennicker, and J. Proença,
Realisability of Global Models of Interaction @ ICTAC'23

Check Realisability and System Synthesis



I. Castellani, M. Mukund, and P.S. Thiagarajan, Synthesizing Distributed Transition Systems from Global Specifications @ FSTTCS'99

cf. our paper for details: M.H. ter Beek, R. Hennicker, and J. Proença, Realisability of Global Models of Interaction @ ICTAC'23

Theorems 2/3

If $RC(\mathcal{M}, \equiv)^{r/p}$ holds, then $\mathcal{M} \sim \otimes^{r/p}_{\Gamma}((\mathcal{M}/\equiv_i)^{r/p})_{i \in I}$

Novelty

1. Realisations of global models with arbitrary multi-interactions supporting any kind of synchronous communication between multiple senders and multiple receivers

Novelty

1. Realisations of global models with arbitrary multi-interactions supporting any kind of synchronous communication between multiple senders and multiple receivers
2. Correctness notion for realisation based on bisimulation rather than isomorhism, so allowing to deal with non-determinism

Novelty

1. Realisations of global models with arbitrary multi-interactions supporting any kind of synchronous communication between multiple senders and multiple receivers
2. Correctness notion for realisation based on bisimulation rather than isomorhism, so allowing to deal with non-determinism
3. To construct realisations we consider, and analyse, two different localisation styles: rich and poor local actions

Novelty

1. Realisations of global models with arbitrary multi-interactions supporting any kind of synchronous communication between multiple senders and multiple receivers
2. Correctness notion for realisation based on bisimulation rather than isomorhism, so allowing to deal with non-determinism
3. To construct realisations we consider, and analyse, two different localisation styles: rich and poor local actions
4. A prototypical tool Ceta checks the realisability conditions and, if they are satisfied, generates local quotients and hence realisations

`https://github.com/arcalab/choreo/tree/ceta`

`https://lmf.di.uminho.pt/ceta`

Online Prototype: `https://lmf.di.uminho.pt/ceta`

Choreographic Extended Team Automata

Choreography

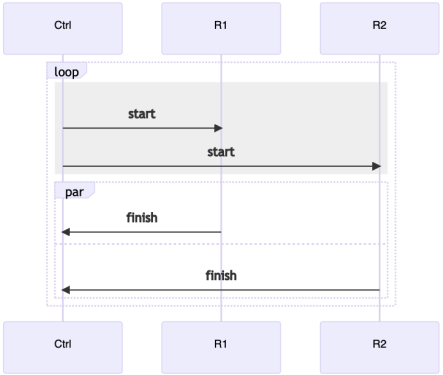
```
1 // Race example
2 (
3  (Ctrl->R1,R2: start);
4  (R1->Ctrl:finish ||
5   R2->Ctrl:finish)
6 )*
```

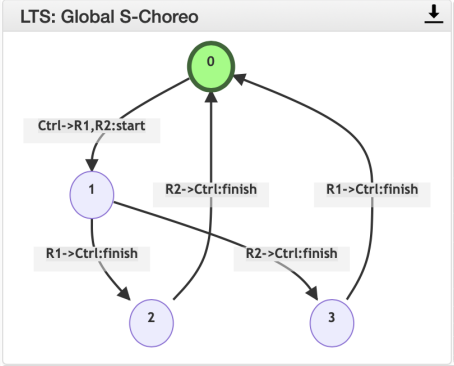
A controller starts 2 runners at the same time, and receives a finish message from each runner at a time.

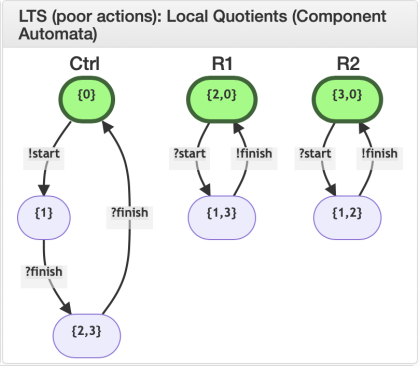
Examples

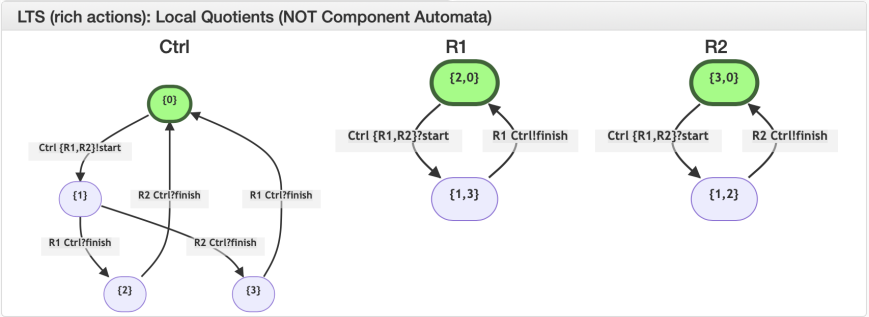
Race (simple) | Race (R1-first) | Race (once, simple) | Toss | Gossip (bad) | Gossip (good) | Cast-v1 | Cast-v2 | ab+cb+ca | ab;ac | ab|ac | ab;cd | ab|cd

Sequence Diagram




LTS: Global S-Choreo
0
Ctrl->R1,R2:start
1
R2->Ctrl:finish
R1->Ctrl:finish
R1->Ctrl:finish
R2->Ctrl:finish
2
3


LTS (poor actions): Local Quotients (Component Automata)
Ctrl
{0}
!start
{1}
?finish
?finish
{2,3}
R1
{2,0}
?start
!finish
{1,3}
R2
{3,0}
?start
!finish
{1,2}


LTS (rich actions): Local Quotients (NOT Component Automata)
Ctrl
{0}
Ctrl {R1,R2}!start
{1}
R2 Ctrl?finish
R1 Ctrl?finish
R1 Ctrl?finish
R2 Ctrl?finish
{2}
{3}
R1
{2,0}
Ctrl {R1,R2}?start
R1 Ctrl!finish
{1,3}
R2
{3,0}
Ctrl {R1,R2}?start
R2 Ctrl!finish
{1,2}

Future Work

Communication properties:

1. Consider final states to distinguish *possible* from *required* communication in case of (local) enabledness of actions (where execution can stop but may also continue, in addition to states where progress *is* required)

Future Work

Communication properties:

1. Consider final states to distinguish *possible* from *required* communication in case of (local) enabledness of actions (where execution can stop but may also continue, in addition to states where progress *is* required)

Realisability:

1. Consider internal actions, using weak bisimulation equivalence for realisations

Future Work

Communication properties:

1. Consider final states to distinguish *possible* from *required* communication in case of (local) enabledness of actions (where execution can stop but may also continue, in addition to states where progress *is* required)

Realisability:

1. Consider internal actions, using weak bisimulation equivalence for realisations
2. Consider open global models (systems) and their composition

Future Work

Communication properties:

1. Consider final states to distinguish *possible* from *required* communication in case of (local) enabledness of actions (where execution can stop but may also continue, in addition to states where progress *is* required)

Realisability:

1. Consider internal actions, using weak bisimulation equivalence for realisations
2. Consider open global models (systems) and their composition
3. Consider realisability conditions in the context of asynchronous communication

Future Work

Communication properties:

1. Consider final states to distinguish *possible* from *required* communication in case of (local) enabledness of actions (where execution can stop but may also continue, in addition to states where progress *is* required)

Realisability:

1. Consider internal actions, using weak bisimulation equivalence for realisations
2. Consider open global models (systems) and their composition
3. Consider realisability conditions in the context of asynchronous communication

Team automata with asynchronous communication?

Thanks for your attention!

Global model $\mathcal{M}$ does not satisfy $\mathbf{RC}(\mathcal{M}, \equiv)$, but $\mathcal{S} = \{\mathcal{M}_p, \mathcal{M}_q, \mathcal{M}_r\}$ does realise $\mathcal{M}$:

global $\mathcal{M}$	local $\mathcal{M}_p$	local $\mathcal{M}_q$	Local $\mathcal{M}_r$	$\mathbf{ta}(\mathcal{S}, \mathbf{st})$