Team Automata: An Overview of Recent Results

Maurice ter Beek

FMT, CNR–ISTI, Pisa, Italy

SySMA Workshop, Lucca, November 8th, 2023

Outline

ICTAC20 Compositionality of Safe Communication in Systems of Team Automata
Extended Team Automata

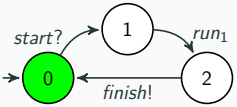
FM21 Featured Team Automata

FM23 Can we Communicate? Using Dynamic Logic to Verify Team Automata
Model Check Team Automata

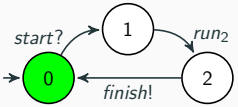
ICTAC23 Realisability of Global Models of Interaction
Realisable Team Automata

Extended Team Automata and other Coordination Models

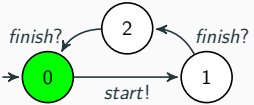
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

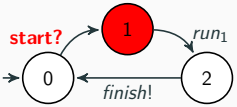


Controller

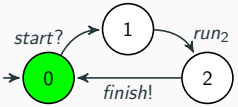
Team Automata

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

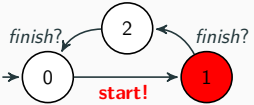
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

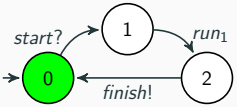


Controller

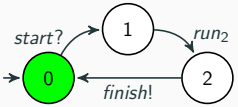
Team Automata

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

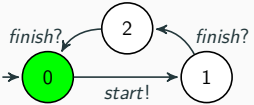
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

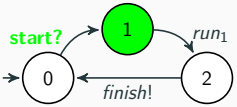


Controller

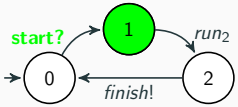
Team Automata

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

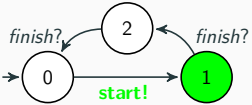
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

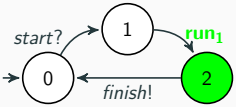


Controller

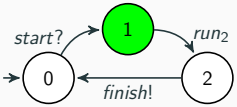
Team Automata

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

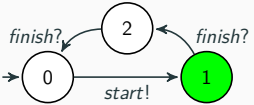
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

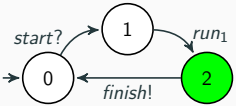


Controller

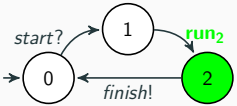
Team Automata

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

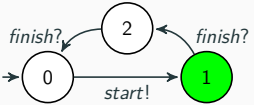
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

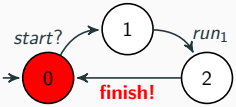


Controller

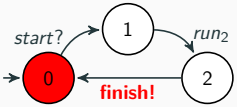
Team Automata

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

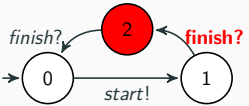
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

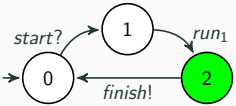


Controller

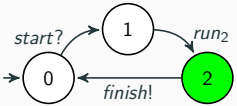
Team Automata

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

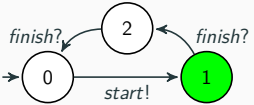
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

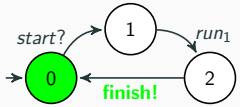


Controller

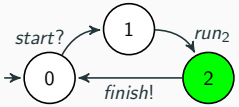
Team Automata

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

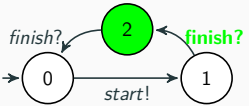
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

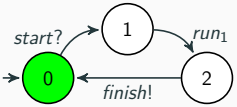


Controller

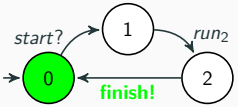
Team Automata

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

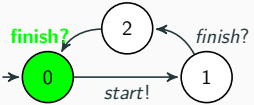
Team Automata: **not all system transitions are meaningful!**



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

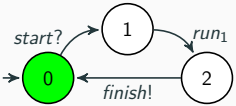


Controller

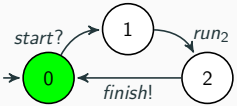
Team Automata

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

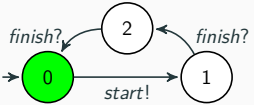
Extended Team Automata: Constrained Multiparty Synchronisations



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$

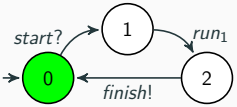


Controller

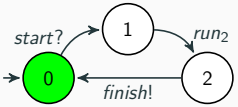
Extended TA synchronisations:

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

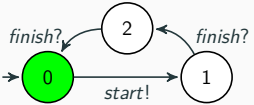
Extended Team Automata: Constrained Multiparty Synchronisations



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



Controller

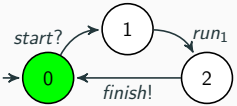
Extended TA synchronisations:

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

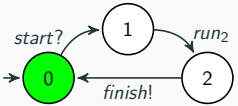
multiparty

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

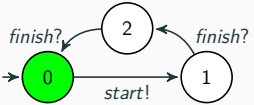
Extended Team Automata: Constrained Multiparty Synchronisations



$\mathcal{R}unner_1$



$\mathcal{R}unner_2$



Controller

Extended TA synchronisations:

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

multiparty

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

constrained

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

Overview on Constrained Multiparty Synchronisation in Team Automata

and other coordination models:

J. Proença @ FACS'23

Overview on Constrained Multiparty Synchronisation in Team Automata

and other coordination models:

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 @ 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 @ 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)

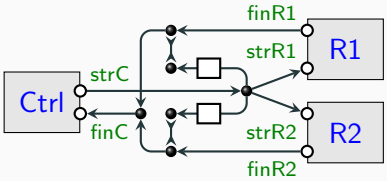
S. Bliudze and J. Sifakis, The algebra of connectors: structuring interaction in BIP. *IEEE Transactions on Computers* 57 (2008)

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

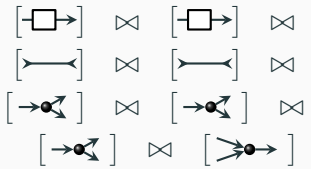
F. Barbanera, I. Lanese, and E. Tuosto, Choreography Automata @ COORDINATION'20

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

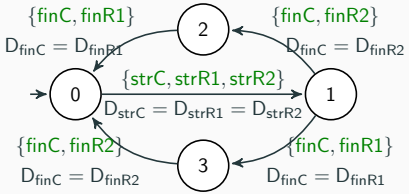
Orchestrators: Reo



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

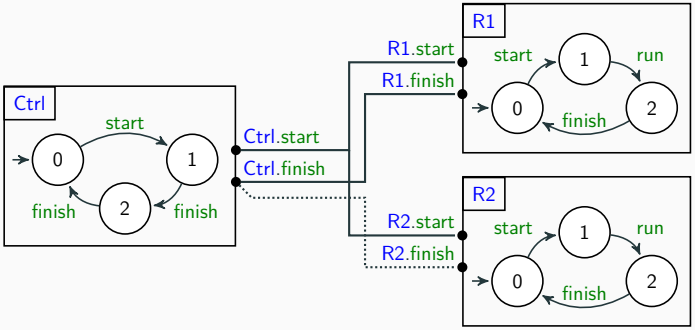


=



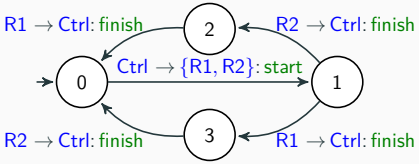
(Semantics with Constraint Automata)

Orchestrators: BIP

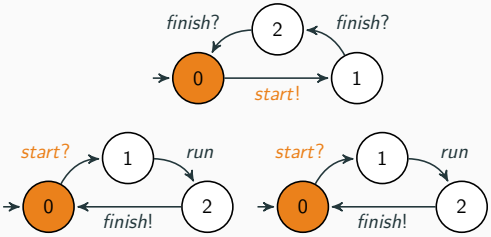


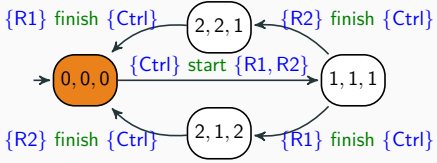
- components expose ports
- interactions restrict which ports can fire
- constructors using unicast (●) and broadcast (▲) can be used to restrict interactions
- dataflow can be added

Choreographies: Choreography Automata

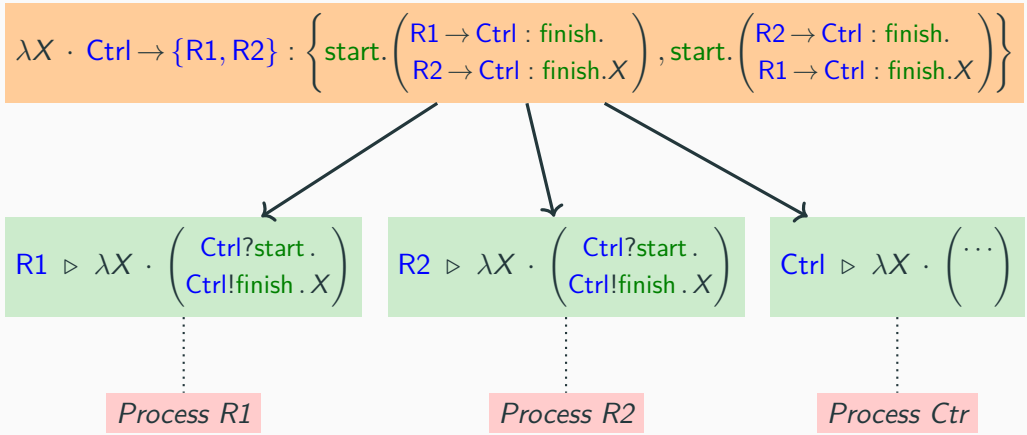


- Many results over the language of CA
- Projections of the language of CA
- Contract automata are similar





(Composed Team Automaton)


$\mathcal{G}$
$\lambda X \cdot \mathsf{Ctrl} \to \{\mathsf{R1}, \mathsf{R2}\} : \left\{ \mathsf{start}. \begin{pmatrix} \mathsf{R1} \to \mathsf{Ctrl} : \mathsf{finish}. \\ \mathsf{R2} \to \mathsf{Ctrl} : \mathsf{finish}.X \end{pmatrix}, \mathsf{start}. \begin{pmatrix} \mathsf{R2} \to \mathsf{Ctrl} : \mathsf{finish}. \\ \mathsf{R1} \to \mathsf{Ctrl} : \mathsf{finish}.X \end{pmatrix} \right\}$
$\mathcal{L}$
$\mathsf{R1} \triangleright \lambda X \cdot \begin{pmatrix} \mathsf{Ctrl?start}\,. \\ \mathsf{Ctrl!finish}\,.\,X \end{pmatrix}$
$\mathsf{R2} \triangleright \lambda X \cdot \begin{pmatrix} \mathsf{Ctrl?start}\,. \\ \mathsf{Ctrl!finish}\,.\,X \end{pmatrix}$
$\mathsf{Ctrl} \triangleright \lambda X \cdot \begin{pmatrix} \cdots \end{pmatrix}$
Process R1
Process R2
Process Ctr

Featured Team Automata

Safe Communication

Extended Team Automata (ETA):

- Synchronisation types per action: *peer-2-peer*, *multicast*, *broadcast*, ...

Goal: safe communication – *no message loss*, *no indefinite waiting*, ...

ter Beek, Hennicker & Kleijn, Compositionality of Safe Communication in Systems of Team Automata @ ICTAC'20

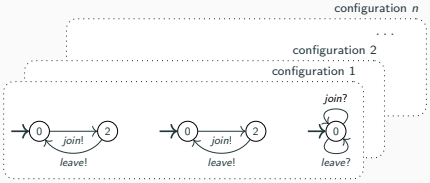
Safe Communication

Extended Team Automata (ETA):

- Synchronisation types per action: *peer-2-peer*, *multicast*, *broadcast*, ...

Goal: safe communication – *no message loss*, *no indefinite waiting*, ...

ter Beek, Hennicker & Kleijn, Compositionality of Safe Communication in Systems of Team Automata @ ICTAC'20



Many systems today are highly configurable (in terms of features):

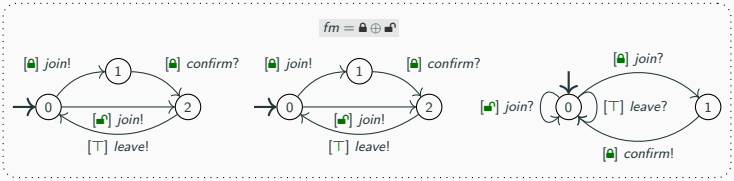
- Large sets of similar systems that share a lot of behaviour but differ in other

Challenge: system-by-system analysis of safe communication quickly becomes unfeasible

Family-based Approach

Featured Team Automata (fETA): ter Beek, Cledou, Hennicker & Proença, *Featured Team Automata* @ FM'21

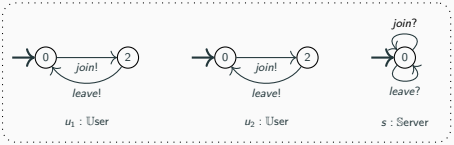
- Families (sets) of Team Automata model as a Software Product Line
- Single model parametrised by features (e.g.: 🔒, 🔓), and a feature model (🔒 ⊕ 🔓)



Goal: family-based analysis of safe communication

Recall: ETA

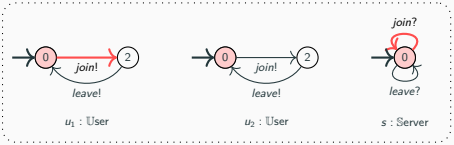
ICTAC'20:



Systems

Recall: ETA

ICTAC'20:

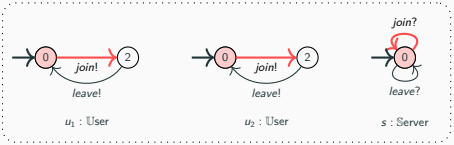


Systems

$$(0,0,0) \xrightarrow{(\{u_1\},join,\{s\})} (2,0,0)$$

Recall: ETA

ICTAC'20:

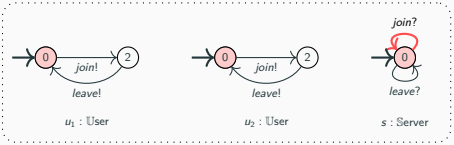


Systems

$$(0,0,0) \xrightarrow{(\{u_1,u_2\},join,\{s\})} (2,2,0)$$

Recall: ETA

ICTAC'20:



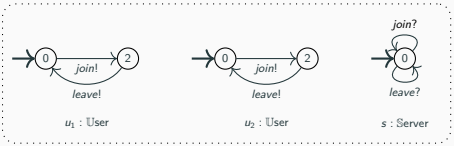
Systems

$(0,0,0) \xrightarrow{(\{\},\, join,\, \{s\})} (0,0,0)$

might not be desirable

Recall: ETA

ICTAC'20:



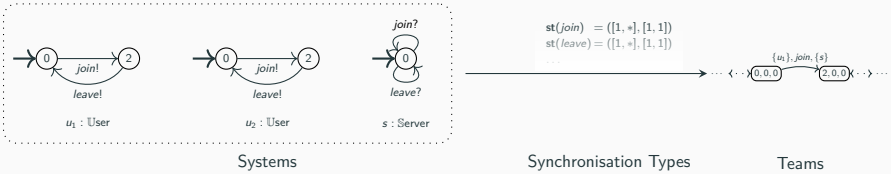
Systems

$\mathbf{st}(join) = ([1, *], [1, 1])$
$\mathbf{st}(leave) = ([1, *], [1, 1])$
...

Synchronisation Types

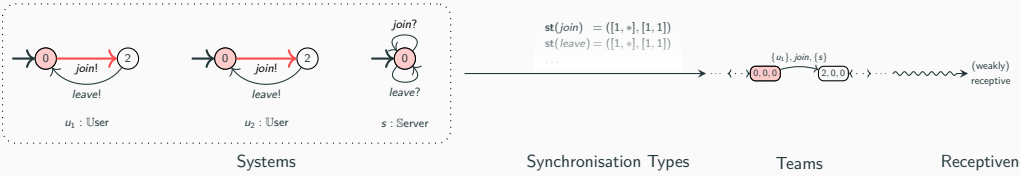
Recall: ETA

ICTAC'20:



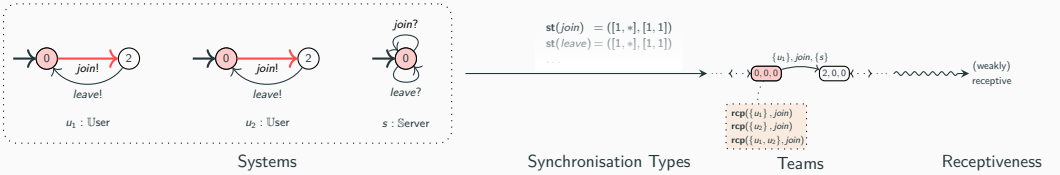
Recall: ETA

ICTAC'20:

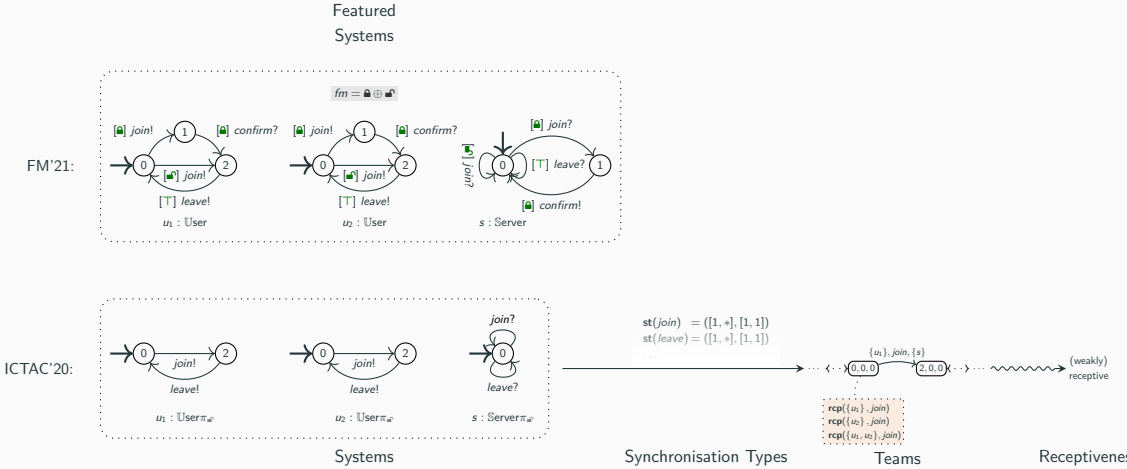


Recall: ETA

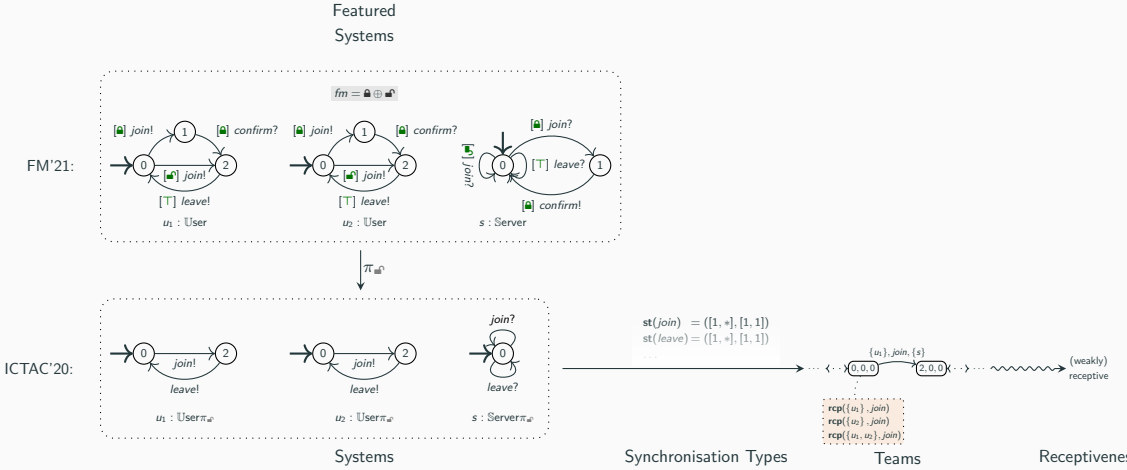
ICTAC'20:



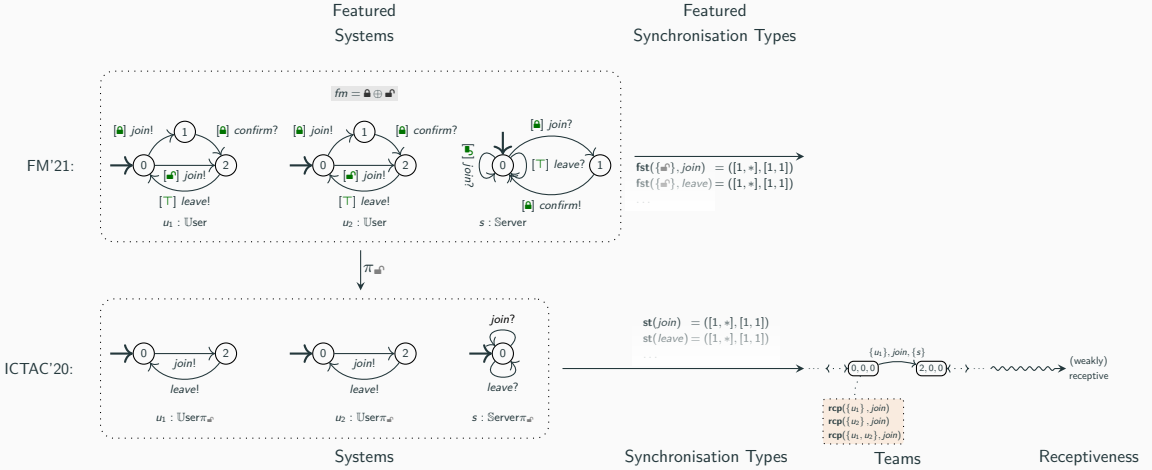
Contributions: fETA



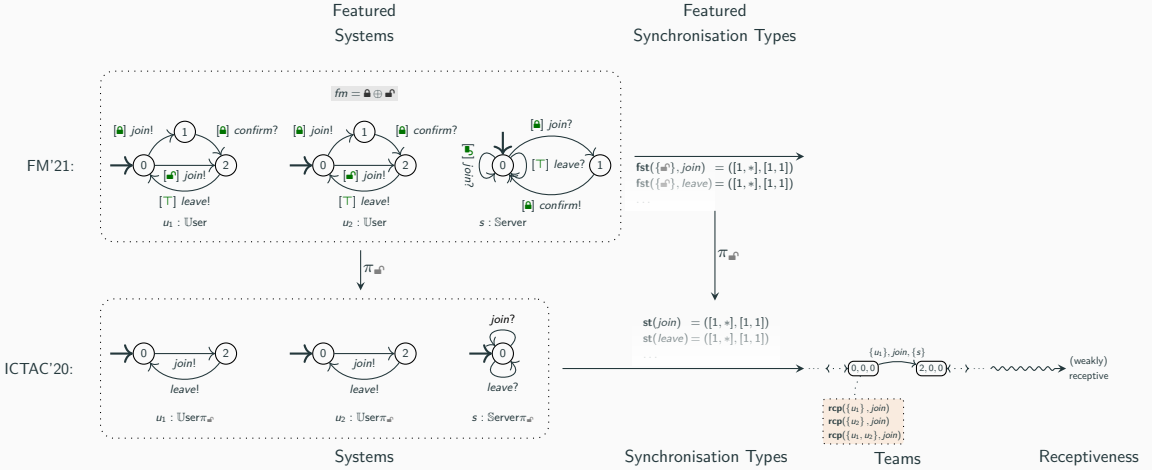
Contributions: fETA



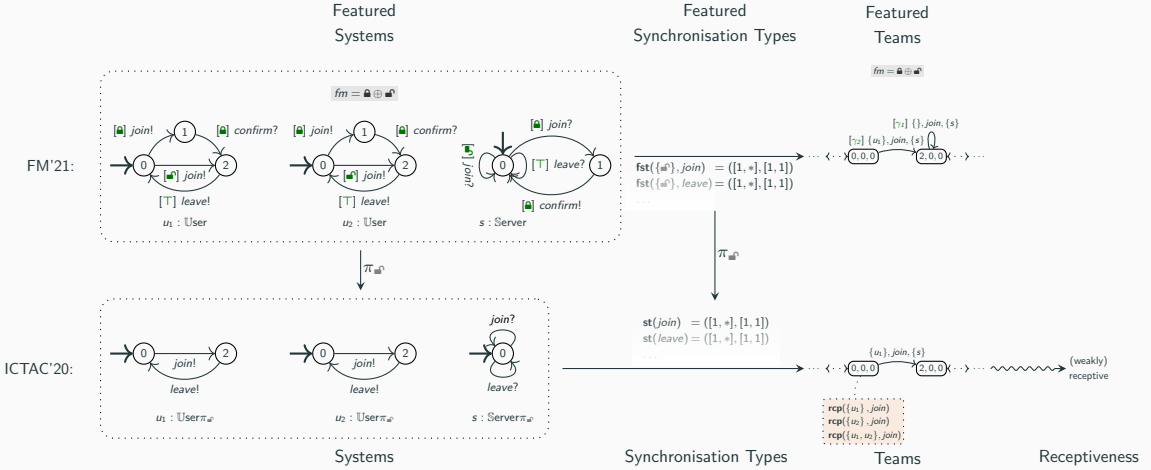
Contributions: fETA



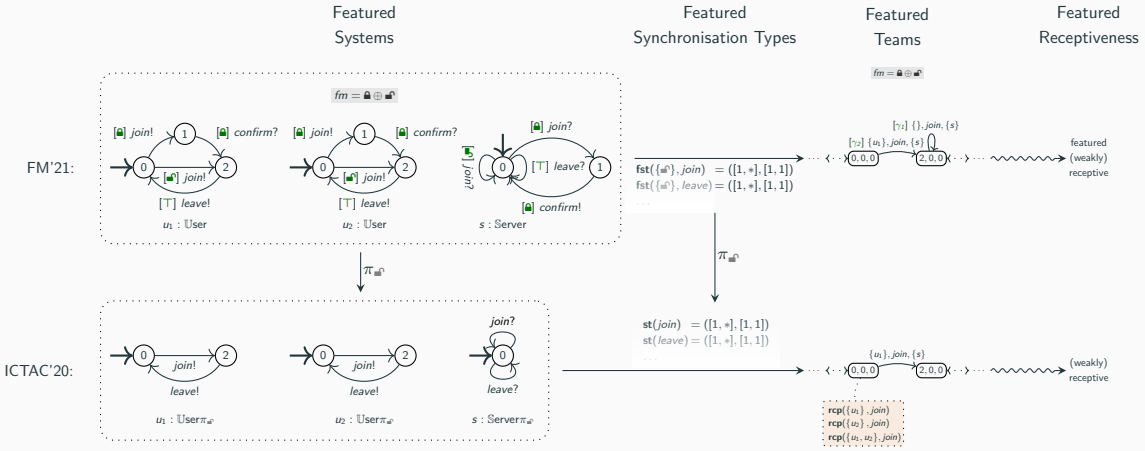
Contributions: fETA



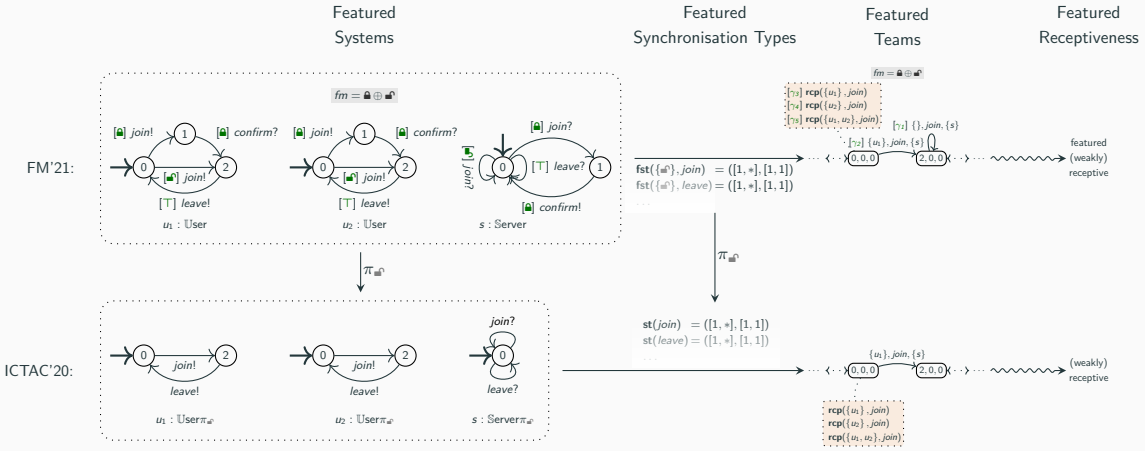
Contributions: fETA



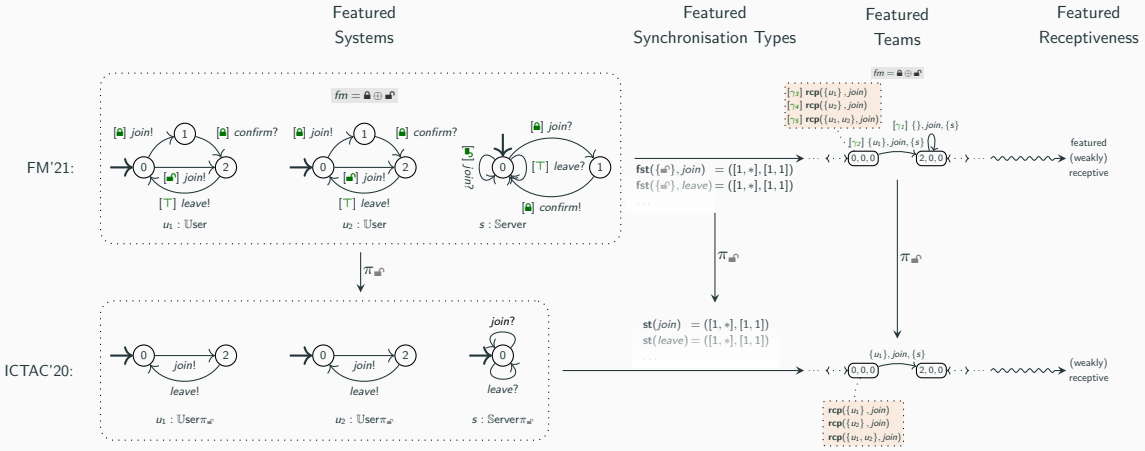
Contributions: fETA



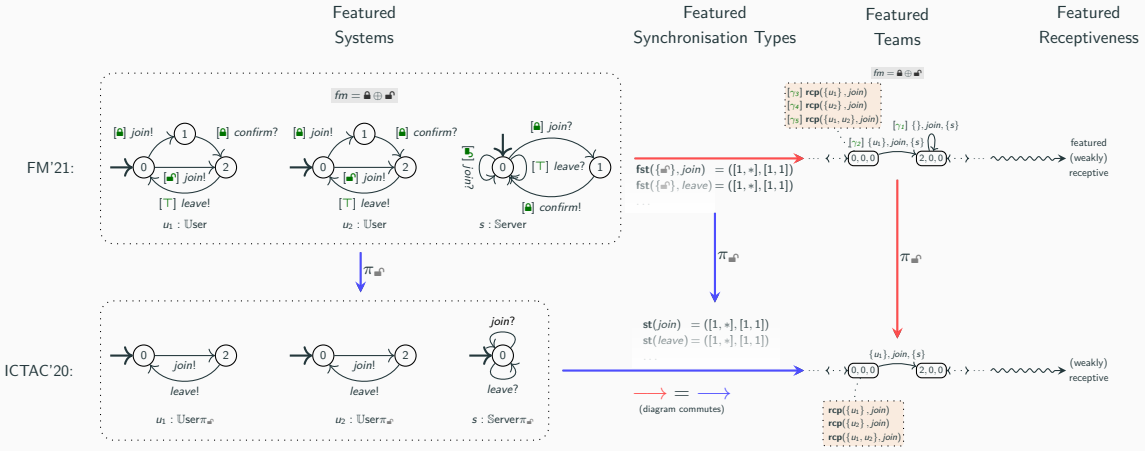
Contributions: fETA



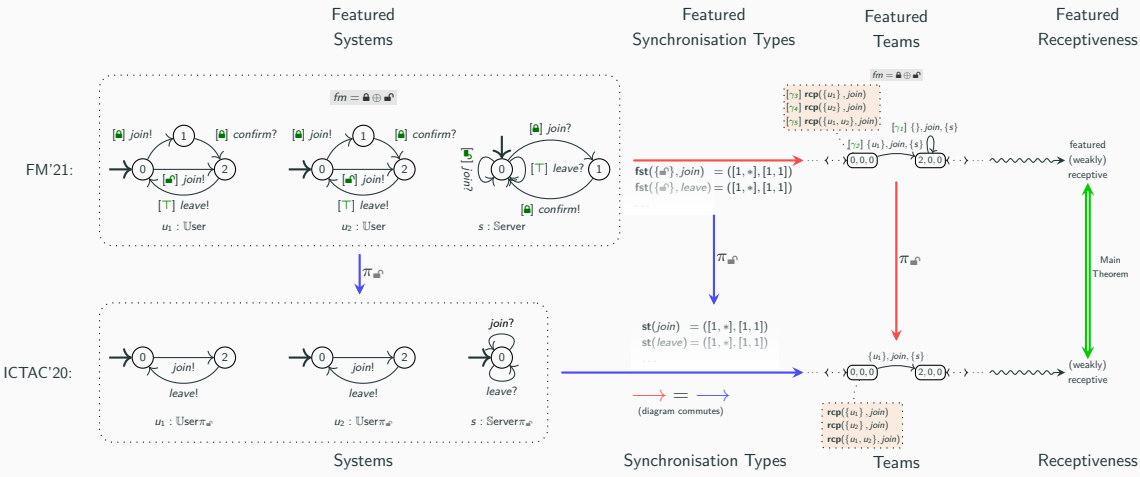
Contributions: fETA



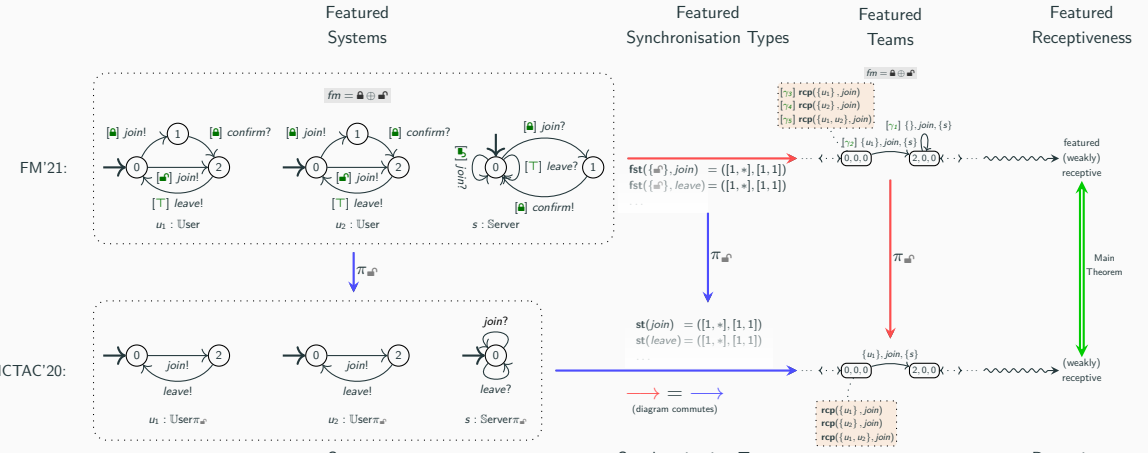
Contributions: fETA



Contributions: fETA



Contributions: fETA

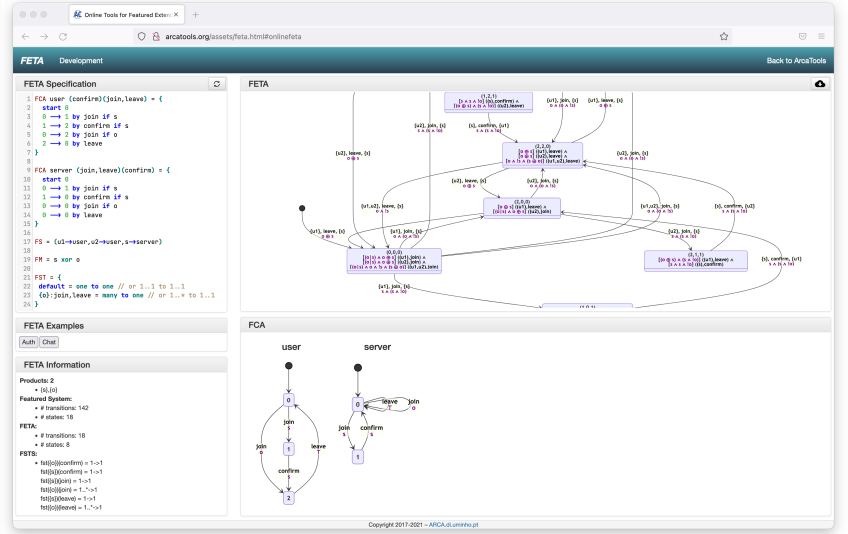


Online prototype: `http://arcatools.org/feta`

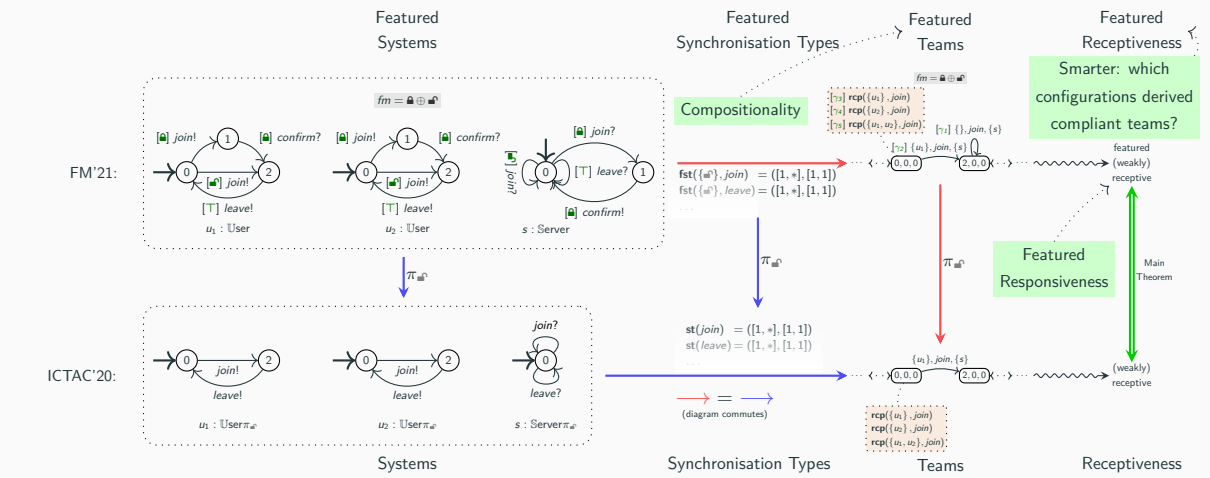
Online Prototype: `http://arcatools.org/feta`

- Specify
- Generate*
- Visualise
- Statistics

*SAT solver to solve *fm*



Future and Ongoing Work



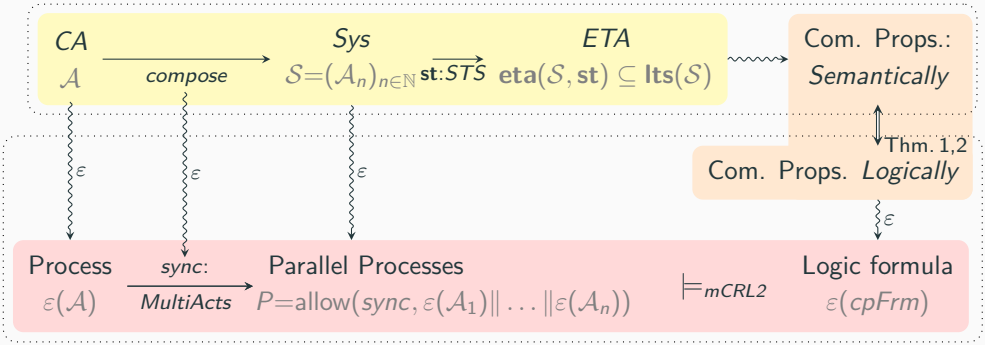
Online prototype: `http://arcatools.org/feta`

Extentions (e.g. FM'23: generate requirements as PDL formulae and check compliance with mCRL2)

Model Check Team Automata

Contributions

ter Beek, Cledou, Hennicker, and Proença, Can we Communicate? Using Dynamic Logic to Verify Team Automata @ FM'23



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

ETA Specification

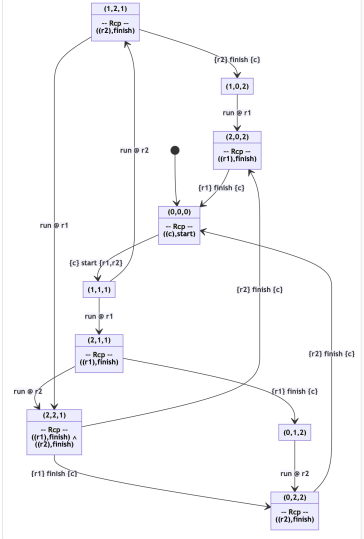
```
1  //Race example
2  CA runner (start)
3           (finish) = {
4    start 0
5    0 --> 1 by start
6    1 --> 2 by run
7    2 --> 0 by finish
8  }
9  CA controller (finish)
10                (start) = {
11   start 0
12   0 --> 1 by start
13   1 --> 2 by finish
14   2 --> 0 by finish
15 }
16 S = (r1:runner, 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

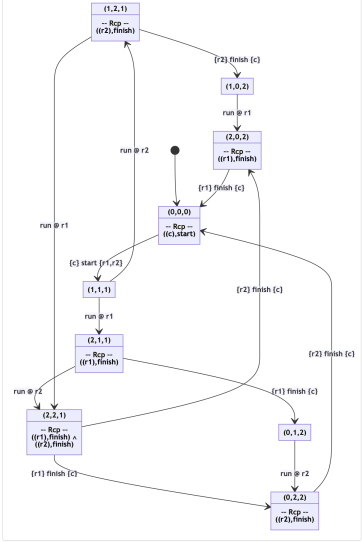
```
1  //Race example
2  CA runner (start)
3           (finish) = {
4    start 0
5    0 --> 1 by start
6    1 --> 2 by run
7    2 --> 0 by finish
8  }
9  CA controller (finish)
10                (start) = {
11   start 0
12   0 --> 1 by start
13   1 --> 2 by finish
14   2 --> 0 by finish
15 }
16 S = (r1:runner,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

Race example

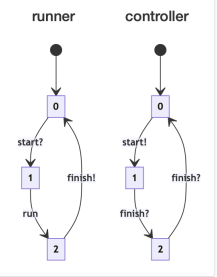
ETA Examples

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:

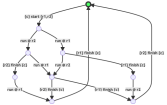
```
[ (r1_finish|c_finish + r2_run + c_start|r1_start|r2_start + r2_finish|c_finish + r1_run)* ](
  ((<r1_finish> true)  =>  (<(r2_run+r2_finish|c_finish)* . r1_finish|c_finish> true)) &&
  ((<r2_finish> true)  =>  (<(r1_finish|c_finish+r1_run)* . r2_finish|c_finish> true)) &&
  ((<c_start> true)  =>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

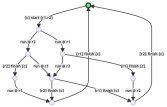
View mCRL2 evidence

Receptiveness: true

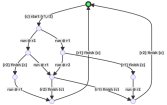


Responsiveness: false

Weak Receptiveness: true



Weak Responsiveness: true



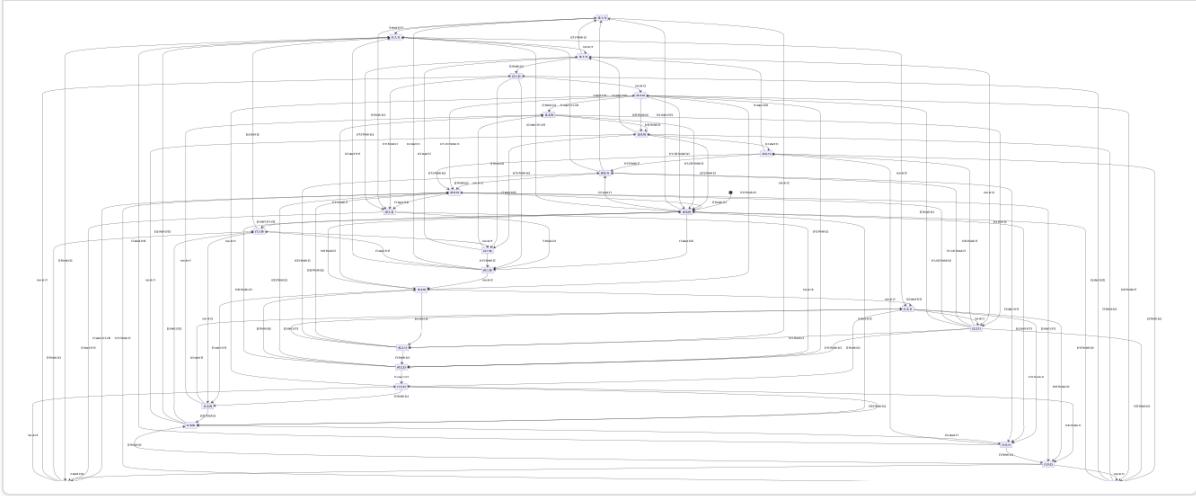
Online Prototype: https://github.com/arcalab/team-a

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

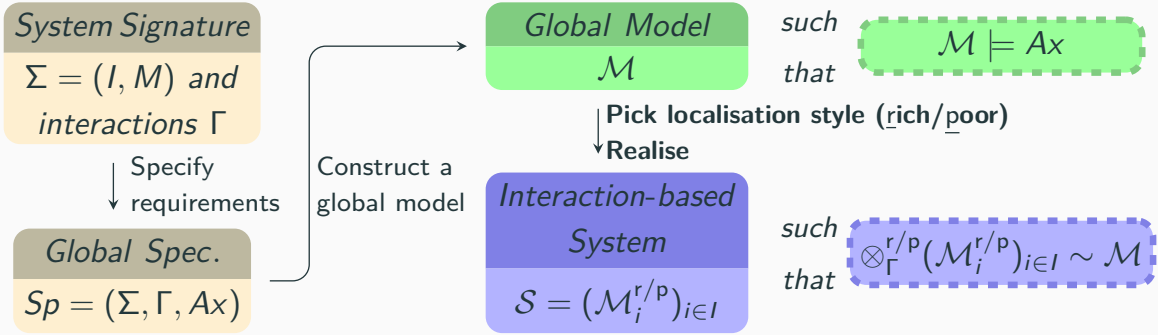
M. Hüttel et al., Foundations of Session Types and Behavioural Contracts. *ACM Comput. Surv.* 49 (2016)

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

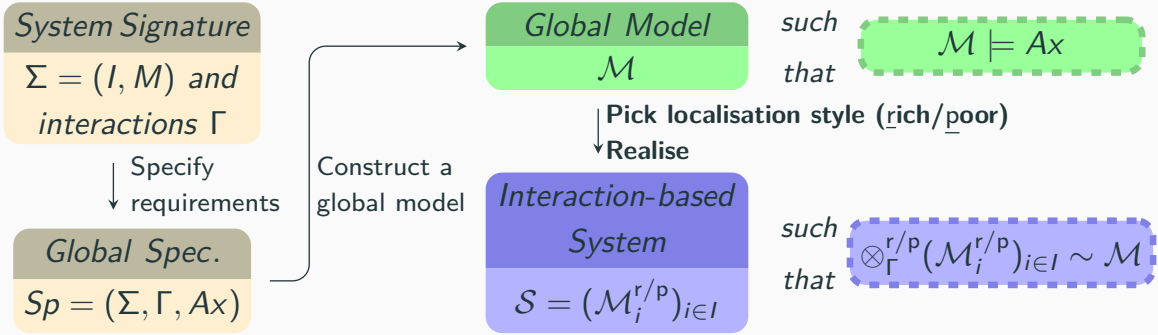
Alternatively, provide a specification in some logical formalism, and construct local models from scratch:

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

Contributions



Contributions



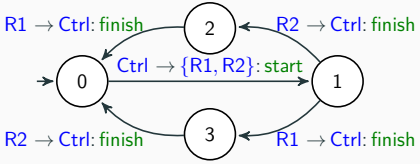
Multi-interactions

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

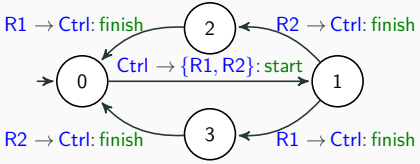
Example

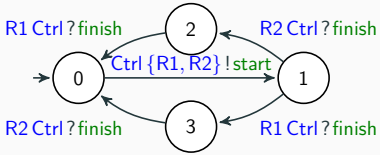
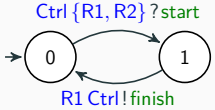
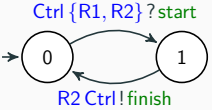
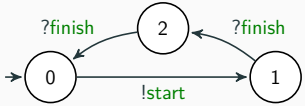
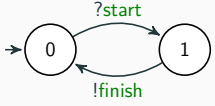
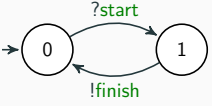
Recall:



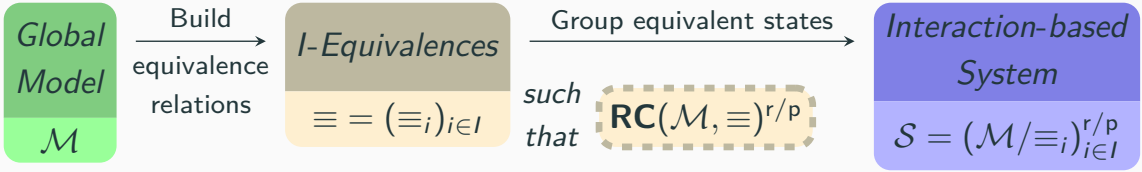
Example

Recall:

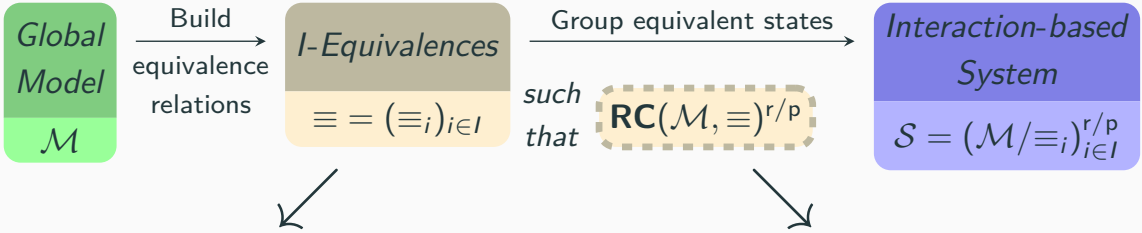


Localisation	Local Ctrl	Local R1	Local R2
Rich			
Poor			

Check Realisability and System Synthesis



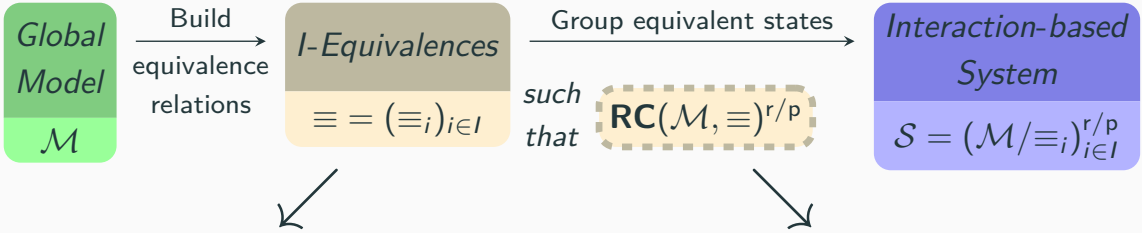
Check Realisability and System Synthesis



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

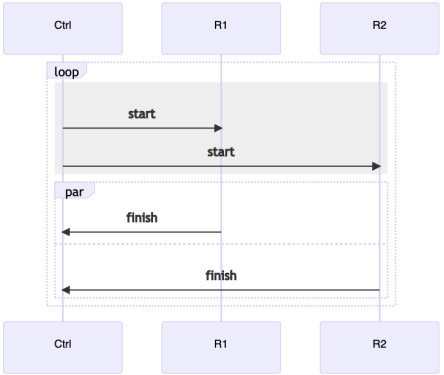
```
1 // Race example
2 (
3  (Ctrl->R1,R2: start);
4  (R1->Ctrl:finish ||
5   R2->Ctrl:finish)
6 )*
```

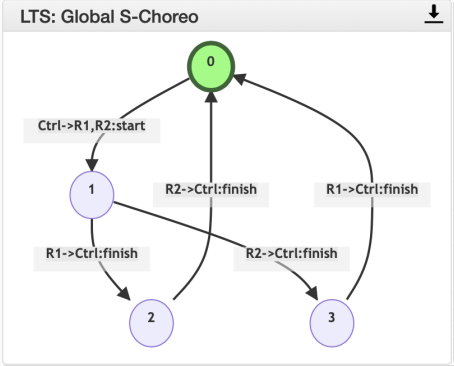
A controller starts 2 runners at the same time, and receives a finish message from each runner at a time.

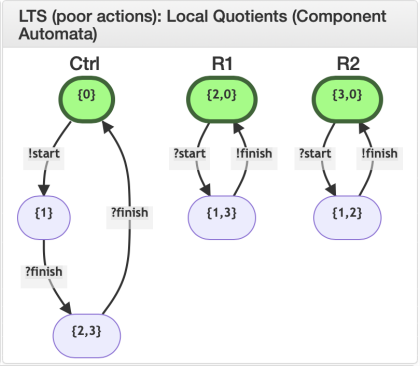
Examples

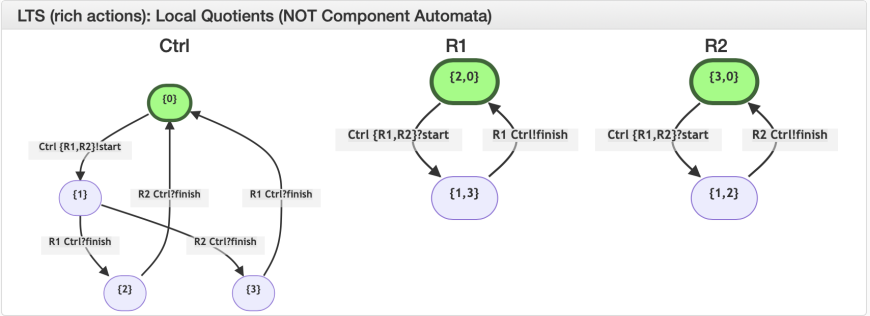
Race (simple) | Race (R1-first) | Race (once, simple) | Toss | Gossip (bad) | Gossip (good) | Cast-v1 | Cast-v2 | ab+cb+ca | ab;ac | ab|ac | ab;cd | ab|cd

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
R1
R2
{0}
{2,0}
{3,0}
!start
?start
!finish
?start
!finish
{1}
{1,3}
{1,2}
?finish
?finish
{2,3}


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

Future Work

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

Future Work

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

2. Consider communication properties, such as receptiveness and responsiveness

M.H. ter Beek, G. Cledou, R. Hennicker, and J. Proença, Can we Communicate? Using Dynamic Logic to Verify Team Automata @ FM'23

M.H. ter Beek, R. Hennicker, and J. Kleijn, Compositionality of Safe Communication in Systems of Team Automata @ ICTAC'20

Future Work

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

2. Consider communication properties, such as receptiveness and responsiveness

M.H. ter Beek, G. Cledou, R. Hennicker, and J. Proença, Can we Communicate? Using Dynamic Logic to Verify Team Automata @ FM'23

M.H. ter Beek, R. Hennicker, and J. Kleijn, Compositionality of Safe Communication in Systems of Team Automata @ ICTAC'20

3. Consider open global models (systems) and their composition

Future Work

1. Consider internal actions, using weak bisimulation equivalence for realisations
2. Consider communication properties, such as receptiveness and responsiveness

M.H. ter Beek, G. Cledou, R. Hennicker, and J. Proença, Can we Communicate? Using Dynamic Logic to Verify Team Automata @ FM'23

M.H. ter Beek, R. Hennicker, and J. Kleijn, Compositionality of Safe Communication in Systems of Team Automata @ ICTAC'20

3. Consider open global models (systems) and their composition
4. Consider realisability conditions in the context of asynchronous communication

Acknowledgements and Publicity

Thanks for your attention! And note that we're hiring!

Thanks also to the other team members of the work presented here:




Thanks for your attention! And note that we're hiring!

Thanks also to the other team members of the work presented here:




And consider submitting your work to FM'24:

- Submission: April 12th
- Conference: September 9th–13th
- Co-located: FMICS, TAP, LOPSTR/PPDP

FM
Milano 2024