Featured Team Automata

Maurice H. ter Beek[1] Guillermina Cledou[2] Rolf Hennicker[3] José Proença[4]

[1]ISTI–CNR, Pisa, Italy

[2]HASLab, INESC TEC & University of Minho, Portugal

[3]Ludwig-Maximilians-Universität München, Munich, Germany

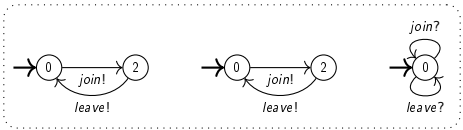
[4]CISTER, ISEP, Polytechnic Institute of Porto, Portugal

FM 2021
25 Nov 2021

Background

Team Automata: [1]

- Systems of communicating components: synchronise over shared actions
- Synchronisation types per action: *peer-2-peer, broadcast, ...*



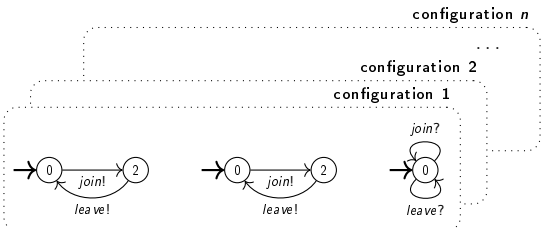
Goal: Safe communication – *no message loss, no indefinite waiting, ...*

[1] ter Beek et al., Compositionality of Safe Communication in Systems of Team Automata. ICTAC 2020

Motivation

Many systems today are highly configurable:

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

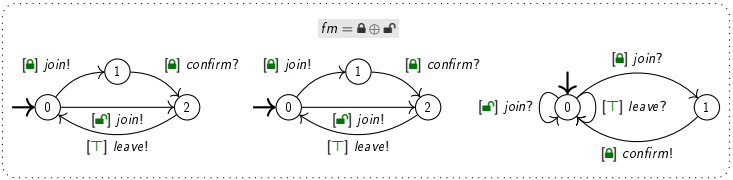


Challenge: System by system analysis of safe communication quickly becomes unfeasible

Approach

Featured Team Automata:

- Families (sets) of Team Automata model as a Software Product Line
- Single model parametrised by features [2] (e.g.: 🔒, 🔓), and a feature model (🔒 ⊕ 🔓)

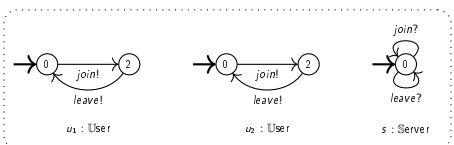


Goal: Family-based analysis of safe communication

[2] Classen et al., Featured Transition Systems: Foundations for Verifying Variability-Intensive Systems and Their Application to LTL Model Checking. IEEE Trans. Softw. Eng. 39 (2013)

Team Automata

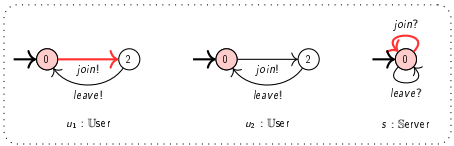
Previous work:



Systems

Team Automata

Previous work:

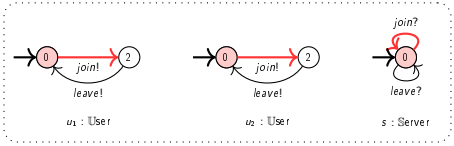


Systems

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

Team Automata

Previous work:

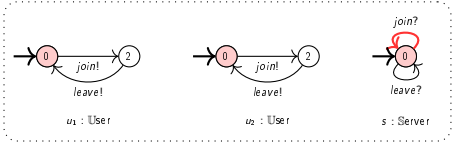


Systems

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

Team Automata

Previous work:



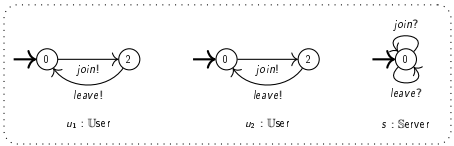
Systems

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

might not be desirable

Team Automata

Previous work:

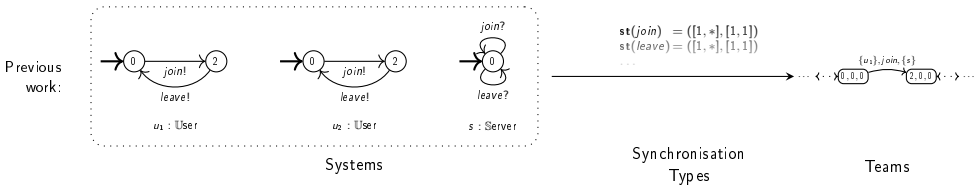


Systems

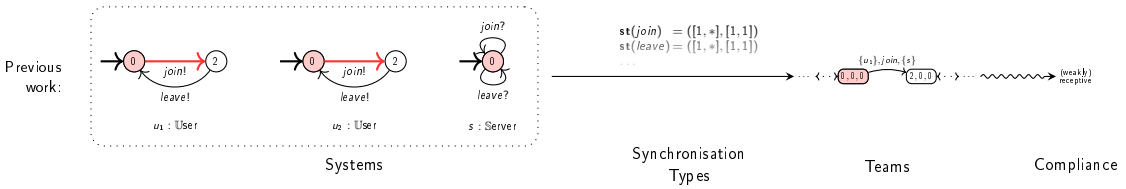
$\mathbf{st}(join) = ([1,*],[1,1])$
$\mathbf{st}(leave) = ([1,*],[1,1])$
...

Synchronisation Types

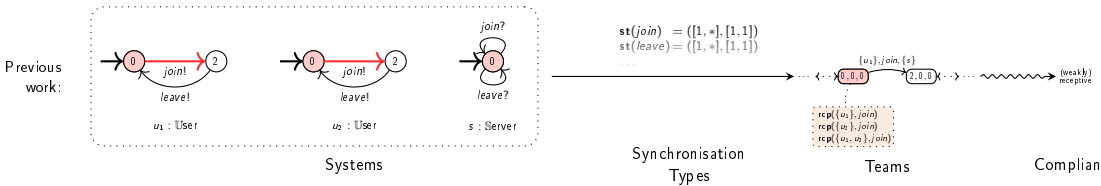
Team Automata



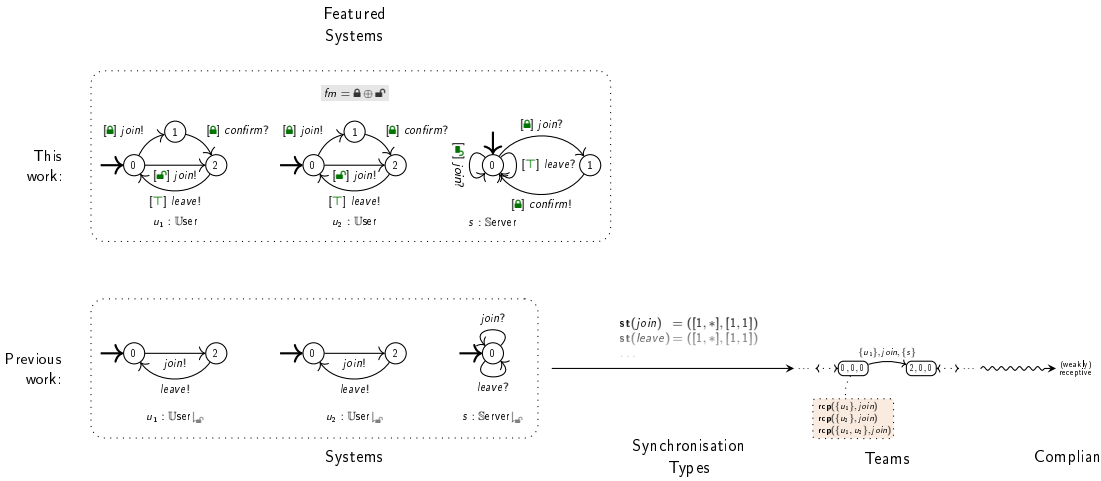
Team Automata



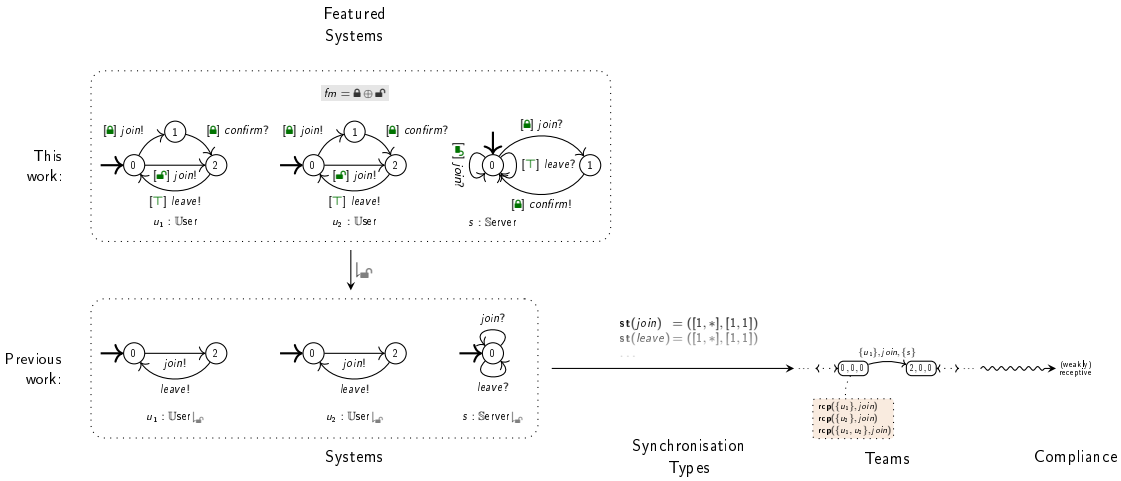
Team Automata



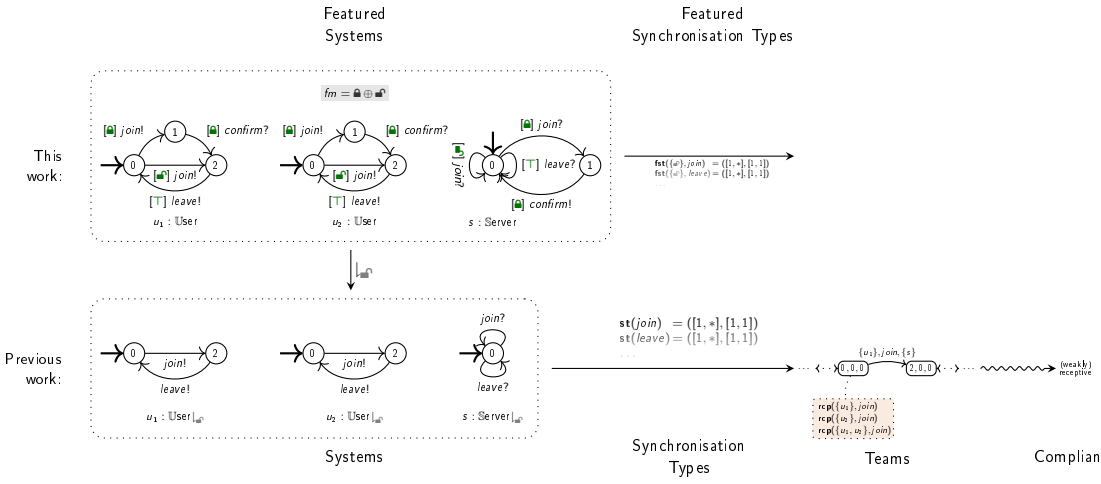
Featured Team Automata



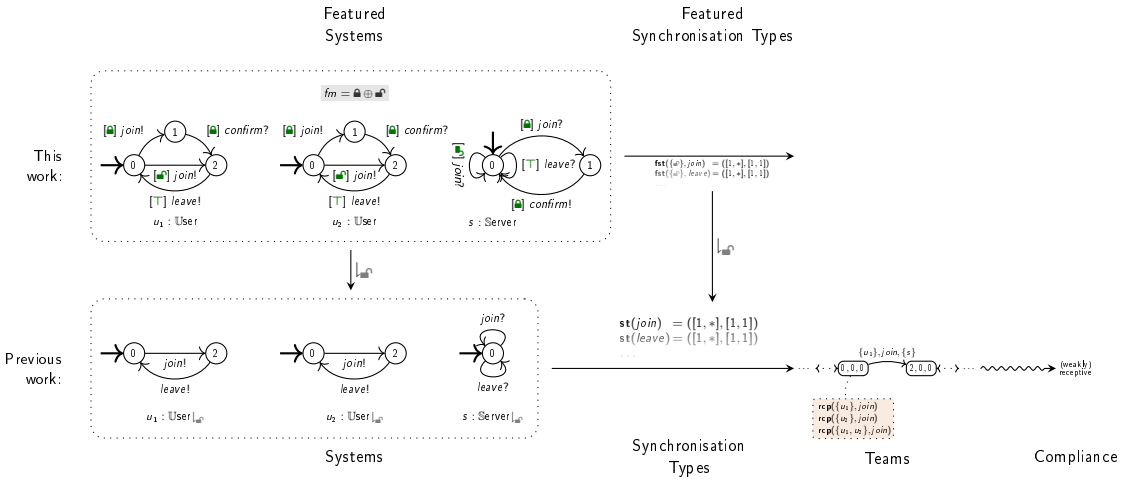
Featured Team Automata



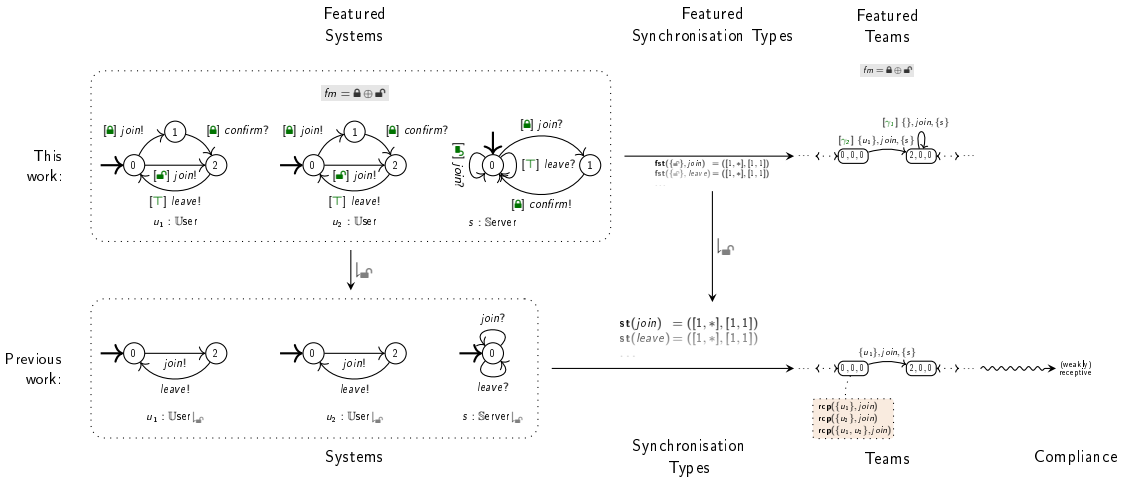
Featured Team Automata



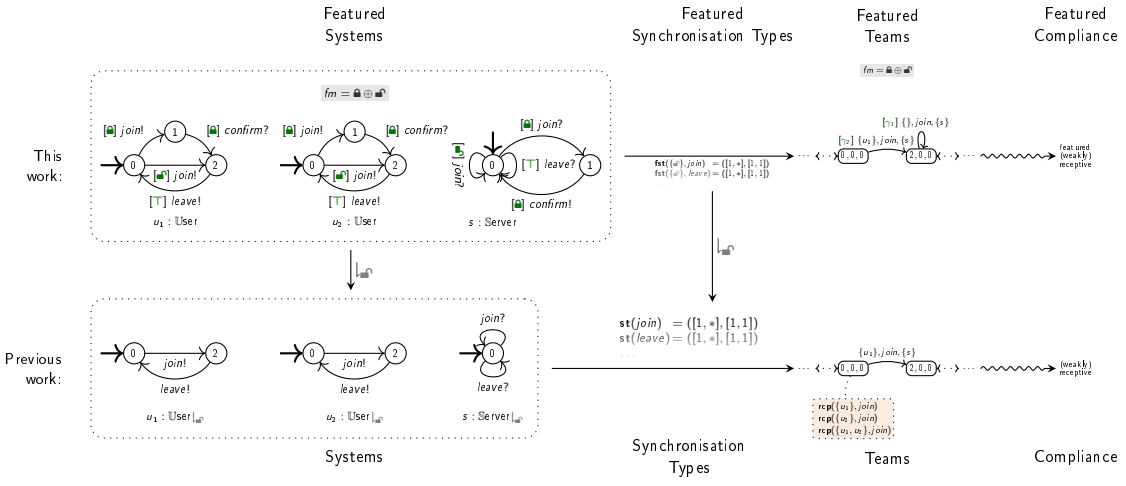
Featured Team Automata



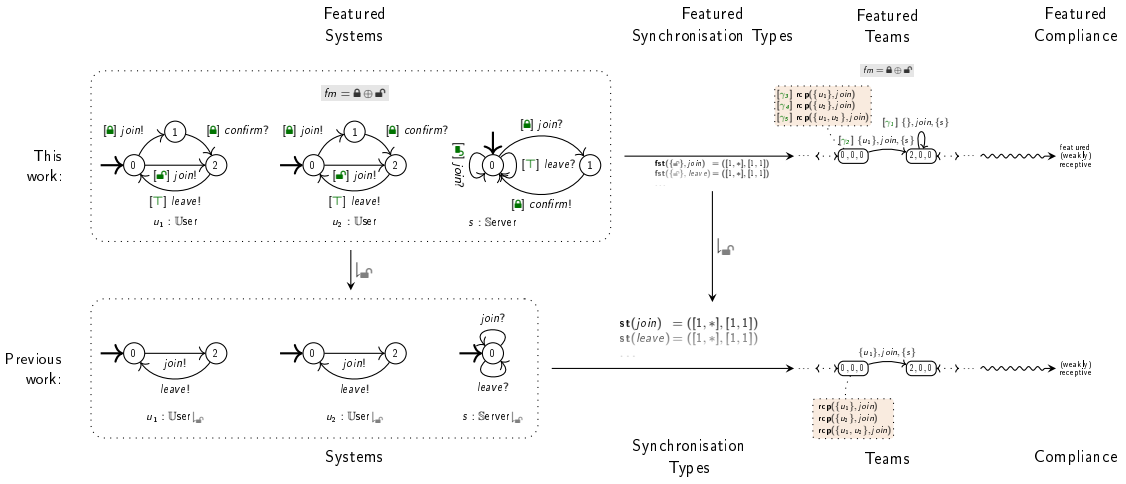
Featured Team Automata



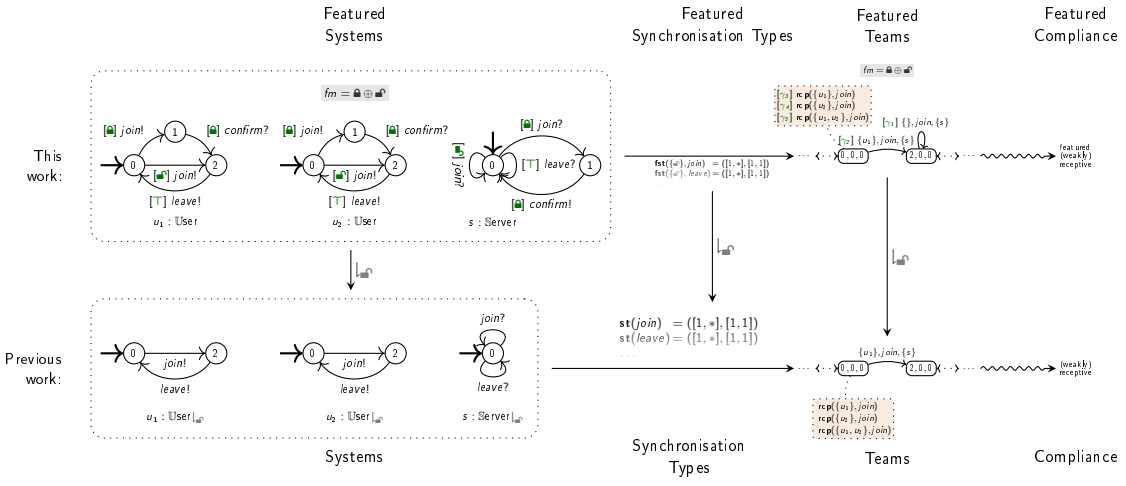
Featured Team Automata



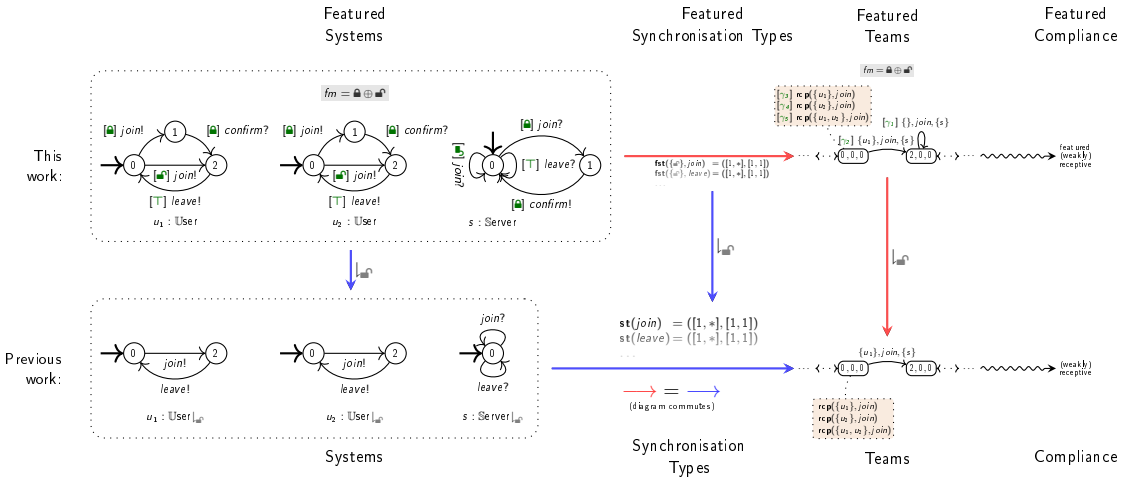
Featured Team Automata



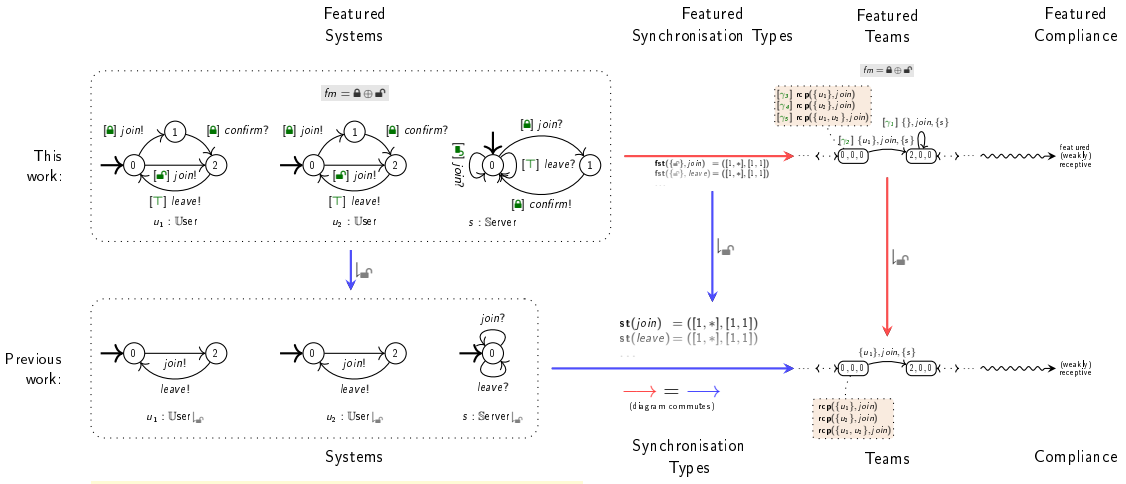
Featured Team Automata



Featured Team Automata

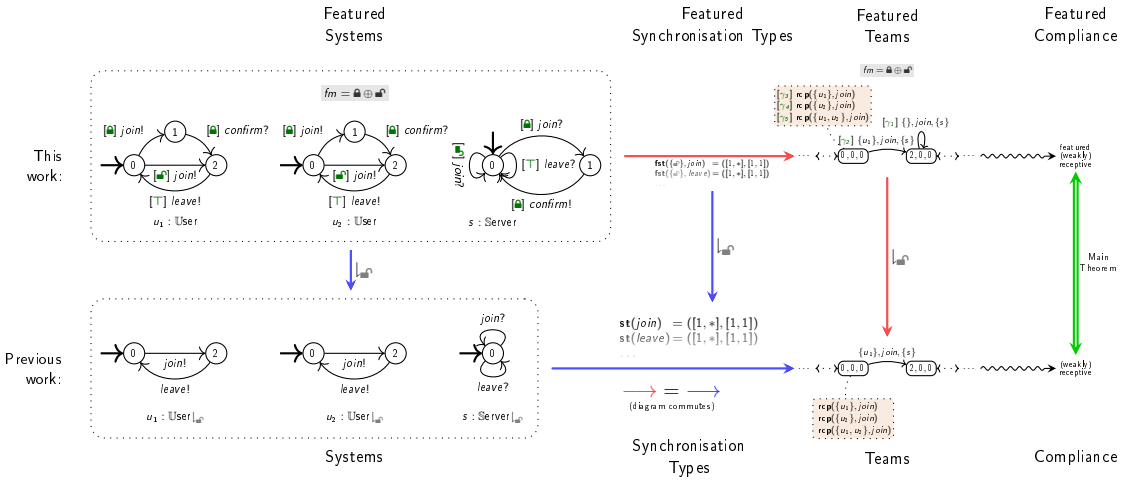


Featured Team Automata



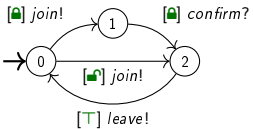
Online prototype: `http://arcatools.org/feta`

Featured Team Automata

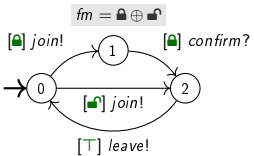


Online prototype: `http://arcatools.org/feta`

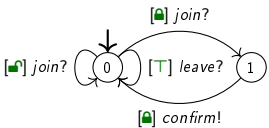
Featured Team Automata Transitions



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



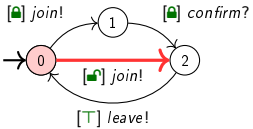
(c) $s : \mathbb{S}\text{erver}$

Transitions are constraint with feature expressions by:

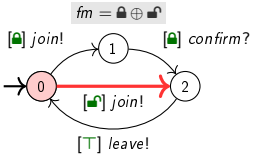
- local feature expressions: characterise the products with all participants present
- **fst**: characterise the products that satisfy the corresponding synchronisation type

$$\textbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \textbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

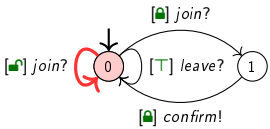
Featured Team Automata Transitions



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

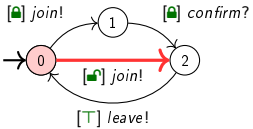
Transitions are constraint with feature expressions by:

- local feature expressions: characterise the products with all participants present
- **fst**: characterise the products that satisfy the corresponding synchronisation type

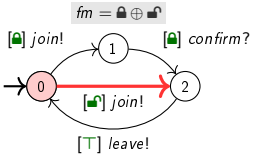
$$\textbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \textbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

$$(0,0,0) \xrightarrow{[\quad](\{u_1,u_2\},join,\{s\})}_{\textbf{fst}[\mathcal{S}]} (2,2,0)$$

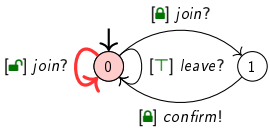
Featured Team Automata Transitions



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

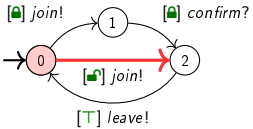
Transitions are constraint with feature expressions by:

- local feature expressions: characterise the products with all participants present
- **fst**: characterise the products that satisfy the corresponding synchronisation type

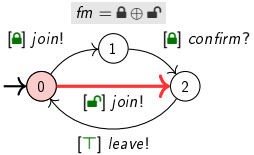
$$\textbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \textbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

$$(0,0,0) \xrightarrow{[🔓\wedge🔓\wedge🔓 \quad](\{u_1,u_2\},join,\{s\})}_{\textbf{fst}[\mathcal{S}]} (2,2,0)$$

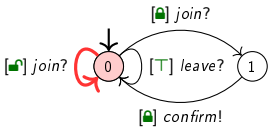
Featured Team Automata Transitions



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

Transitions are constraint with feature expressions by:

- local feature expressions: characterise the products with all participants present
- **fst**: characterise the products that satisfy the corresponding synchronisation type

$$\textbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \textbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

$$(0,0,0) \xrightarrow{[🔓\wedge🔓\wedge🔓\wedge🔓\wedge\neg🔒](\{u_1,u_2\},join,\{s\})}_{\textbf{fst}[\mathcal{S}]} (2,2,0)$$

Featured Receptiveness Requirements

Here: Receptiveness – no message loss

Featured Receptiveness Requirements

Here: Receptiveness – no message loss

Featured Receptiveness Requirement:

Whenever (a group of) components want to send a message (in a product), there should be (a group of) components (in the same product) ready to receive the message in conformance with the synchronisation type

Featured Receptiveness Requirements

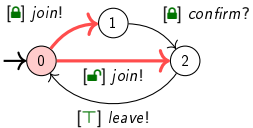
Here: Receptiveness – no message loss

Featured Receptiveness Requirement:

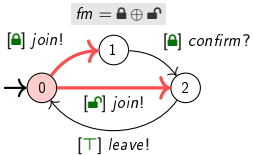
Whenever (a group of) components want to send a message (in a product), there should be (a group of) components (in the same product) ready to receive the message in conformance with the synchronisation type

A Featured Team Automata is (weakly) receptive, if it is (weakly) compliant with all its featured requirements

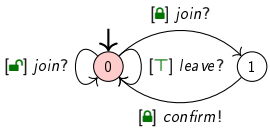
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



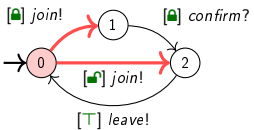
(c) $s : \mathbb{S}\text{erver}$

$$\mathbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \mathbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

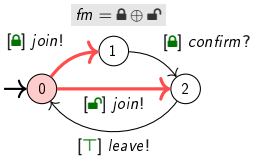
At state $(0,0,0)$:

[ ] $\mathbf{rcp}(\{u_1\}, join)$ $\wedge$ $\mathbf{rcp}(\{u_2\}, join)$ $\wedge$ [ ] $\mathbf{rcp}(\{u_1, u_2\}, join)$

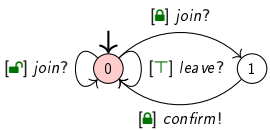
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

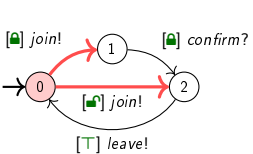
$$\mathbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \mathbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

At state $(0,0,0)$:

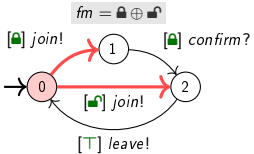
$$[\quad\quad] \mathbf{rcp}(\{u_1\}, join) \wedge \mathbf{rcp}(\{u_2\}, join) \wedge [\quad\quad] \mathbf{rcp}(\{u_1, u_2\}, join)$$

Featured receptiveness requirements are constraint with feature expression by:

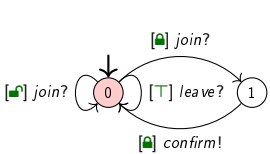
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\mathbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \mathbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

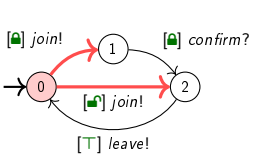
At state $(0,0,0)$:

[ ] $\mathbf{rcp}(\{u_1\}, join) \wedge \mathbf{rcp}(\{u_2\}, join) \wedge$ [ ] $\mathbf{rcp}(\{u_1, u_2\}, join)$

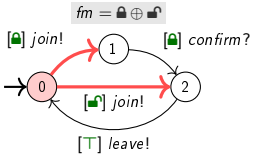
Featured receptiveness requirements are constraint with feature expression by:

- local feature expressions: characterise products where the participants are present

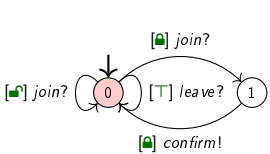
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\mathbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \mathbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

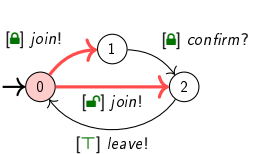
At state $(0,0,0)$:

$[🔒 \vee 🔓 \quad]\ \mathbf{rcp}(\{u_1\}, join)\ \wedge \quad \mathbf{rcp}(\{u_2\}, join)\ \wedge\ [\quad]\ \mathbf{rcp}(\{u_1, u_2\}, join)$

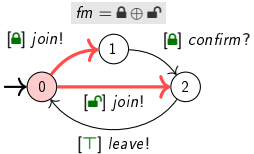
Featured receptiveness requirements are constraint with feature expression by:

- local feature expressions: characterise products where the participants are present

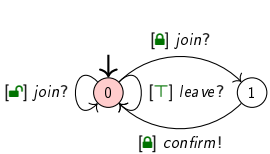
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\textbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \textbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

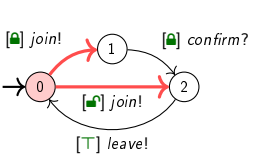
At state $(0,0,0)$:

[🔒 ∨ 🔓] $\textbf{rcp}(\{u_1\}, join) \wedge$ $\textbf{rcp}(\{u_2\}, join) \wedge$ [ ] $\textbf{rcp}(\{u_1, u_2\}, join)$

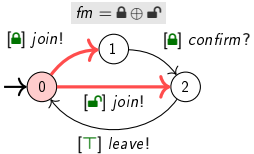
Featured receptiveness requirements are constraint with feature expression by:

- local feature expressions: characterise products where the participants are present
- **fst**: characterise products with the correct number of senders

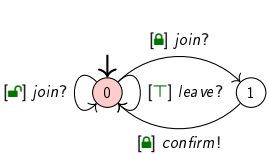
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\mathbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \mathbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

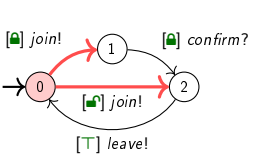
At state $(0,0,0)$:

$[🔒 \vee 🔓 \wedge fm]\ \mathbf{rcp}(\{u_1\}, join)\ \wedge \qquad \mathbf{rcp}(\{u_2\}, join)\ \wedge\ [\qquad\qquad]\ \mathbf{rcp}(\{u_1, u_2\}, join)$

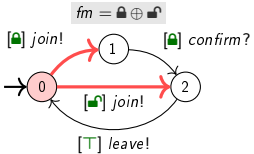
Featured receptiveness requirements are constraint with feature expression by:

- local feature expressions: characterise products where the participants are present
- **fst**: characterise products with the correct number of senders

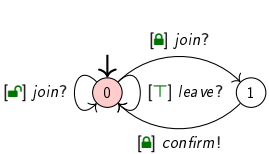
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\mathbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \mathbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

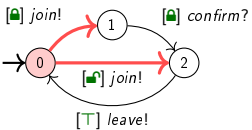
At state $(0,0,0)$:

$[🔒 \vee 🔓 \wedge fm]\ \mathbf{rcp}(\{u_1\}, join)\ \wedge \qquad \mathbf{rcp}(\{u_2\}, join)\ \wedge\ [\qquad\qquad]\ \mathbf{rcp}(\{u_1, u_2\}, join)$

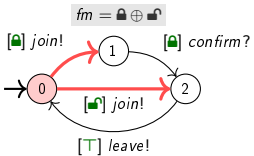
Featured receptiveness requirements are constraint with feature expression by:

- local feature expressions: characterise products where the participants are present
- **fst**: characterise products with the correct number of senders
- reachable states: characterise products where the state is reachable

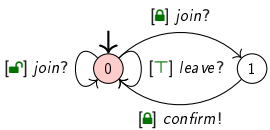
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\mathbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \mathbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

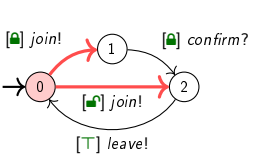
At state $(0,0,0)$:

$[🔒 \vee 🔓 \wedge fm]\ \mathbf{rcp}(\{u_1\}, join)\ \wedge\ [fm]\ \mathbf{rcp}(\{u_2\}, join)\ \wedge\ [\qquad\qquad]\ \mathbf{rcp}(\{u_1, u_2\}, join)$

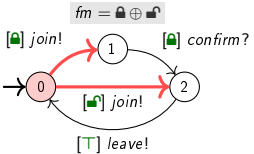
Featured receptiveness requirements are constraint with feature expression by:

- local feature expressions: characterise products where the participants are present
- **fst**: characterise products with the correct number of senders
- reachable states: characterise products where the state is reachable

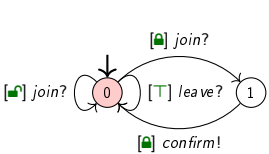
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\textbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \textbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

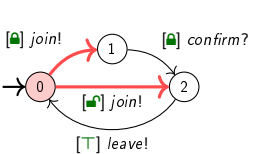
At state $(0,0,0)$:

$$[🔒 \vee 🔓 \wedge fm]\ \textbf{rcp}(\{u_1\}, join)\ \wedge\ [fm]\ \textbf{rcp}(\{u_2\}, join)\ \wedge\ [🔒 \vee 🔓 \qquad]\ \textbf{rcp}(\{u_1, u_2\}, join)$$

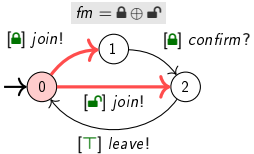
Featured receptiveness requirements are constraint with feature expression by:

- local feature expressions: characterise products where the participants are present
- **fst**: characterise products with the correct number of senders
- reachable states: characterise products where the state is reachable

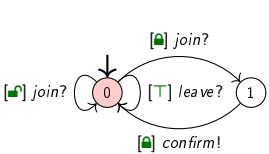
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\textbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \textbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

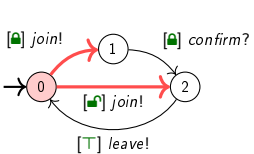
At state $(0,0,0)$:

$$[🔒 \vee 🔓 \wedge fm]\ \textbf{rcp}(\{u_1\}, join)\ \wedge\ [fm]\ \textbf{rcp}(\{u_2\}, join)\ \wedge\ [🔒 \vee 🔓 \wedge 🔓 \wedge \neg 🔒 \quad]\ \textbf{rcp}(\{u_1, u_2\}, join)$$

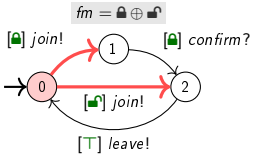
Featured receptiveness requirements are constraint with feature expression by:

- local feature expressions: characterise products where the participants are present
- **fst**: characterise products with the correct number of senders
- reachable states: characterise products where the state is reachable

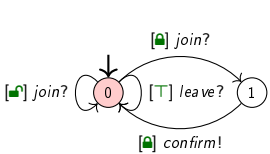
Featured Receptiveness Requirements



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $s : \mathbb{S}\text{erver}$

$$\textbf{fst}(\{🔒\}, join) = ([1,1],[1,1]) \qquad \textbf{fst}(\{🔓\}, join) = ([1,*],[1,1])$$

At state $(0,0,0)$:

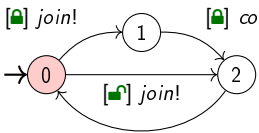
$$[🔒 \vee 🔓 \wedge fm]\ \textbf{rcp}(\{u_1\}, join)\ \wedge\ [fm]\ \textbf{rcp}(\{u_2\}, join)\ \wedge\ [🔒 \vee 🔓 \wedge 🔓 \wedge \neg 🔒 \wedge fm]\ \textbf{rcp}(\{u_1, u_2\}, join)$$

Featured receptiveness requirements are constraint with feature expression by:

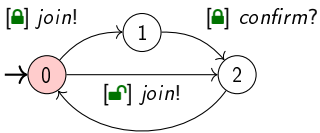
- local feature expressions: characterise products where the participants are present
- **fst**: characterise products with the correct number of senders
- reachable states: characterise products where the state is reachable

Compliance with requirements

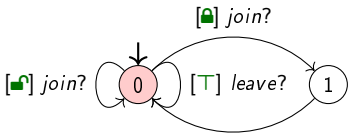
$fm = 🔒 \oplus 🔓$



(a) $u_1 : \mathbb{U}\text{ser}$



(b) $u_2 : \mathbb{U}\text{ser}$



(c) $\mathbb{S}\text{erver}$

At state $(0,0,0)$:

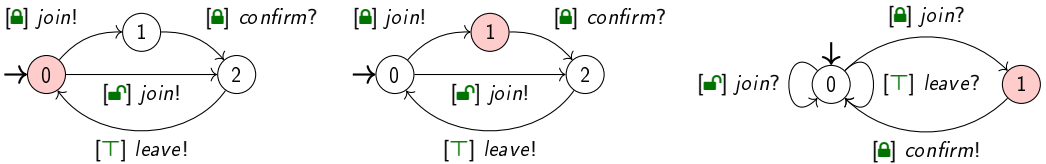
$$\underline{[fm]\ \mathbf{rcp}(\{u_1\}, join)} \ \wedge\ [fm]\ \mathbf{rcp}(\{u_2\}, join) \ \wedge\ [🔓 \wedge \neg 🔒]\ \mathbf{rcp}(\{u_1, u_2\}, join)$$

$$\{🔒\} : (0,0,0) \xrightarrow{[🔒 \wedge fm]\ (\{u_1\}, join, \{s\})}_{\mathbf{fst}[\mathcal{S}]} (1,0,1)$$

$$\{🔓\} : (0,0,0) \xrightarrow{[🔓 \wedge fm]\ (\{u_1\}, join, \{s\})}_{\mathbf{fst}[\mathcal{S}]} (2,0,0)$$

Compliance with requirements

$fm = 🔒 \oplus 🔓$



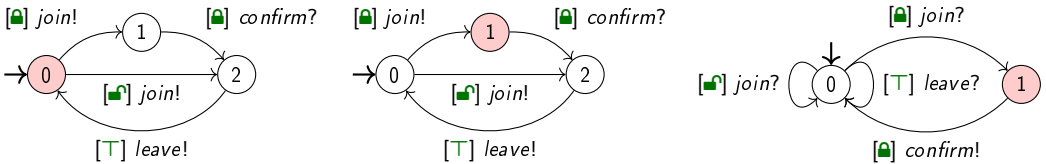
(a) $u_1 : \mathbb{U}\text{ser}$ (b) $u_2 : \mathbb{U}\text{ser}$ (c) $\mathbb{S}\text{erver}$

At state $(0, 1, 1)$:

$$\underline{[🔒 \wedge \neg 🔓]\ \mathbf{rcp}(\{u_1\}, join)} \ \wedge \ [🔒 \wedge \neg 🔓]\ \mathbf{rcp}(\{s\}, confirm)$$

Compliance with requirements

$fm = 🔒 \oplus 🔓$



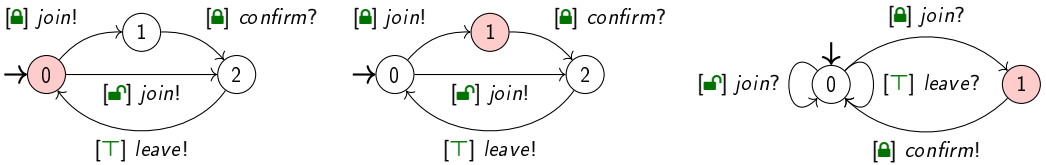
(a) $u_1 : \mathbb{U}\text{ser}$ (b) $u_2 : \mathbb{U}\text{ser}$ (c) $\mathbb{S}\text{erver}$

At state $(0, 1, 1)$:

$$✗\underline{[🔒 \wedge \neg 🔓]\ \mathbf{rcp}(\{u_1\}, join)} \ \wedge\ [🔒 \wedge \neg 🔓]\ \mathbf{rcp}(\{s\}, confirm)$$

Weak compliance with requirements

$fm = 🔒 \oplus 🔓$



(a) $u_1 : \mathbb{U}\text{ser}$ (b) $u_2 : \mathbb{U}\text{ser}$ (c) $\mathbb{S}\text{erver}$

At state $(0, 1, 1)$:

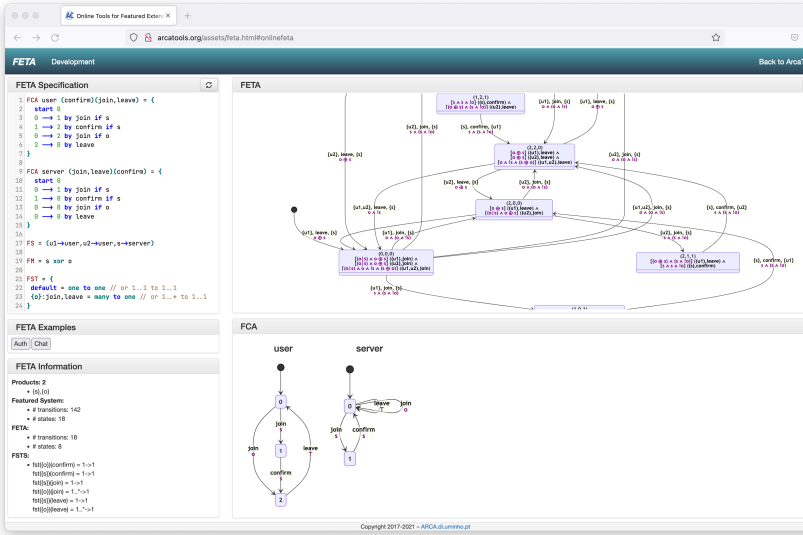
$$\underline{[🔒 \wedge \neg 🔓]\ \mathbf{rcp}(\{u_1\}, join)} \ \wedge\ [🔒 \wedge \neg 🔓]\ \mathbf{rcp}(\{s\}, confirm)$$

$$\{🔒\} :\ (0,1,1) \xrightarrow{[🔒 \wedge \underline{fm}](\{s\}, confirm, \{u_2\})}_{\mathbf{fst}[\mathcal{S}]} (0,2,0) \xrightarrow{[🔒 \wedge \underline{fm}](\{u_1\}, join, \{s\})}_{\mathbf{fst}[\mathcal{S}]} (1,2,1)$$

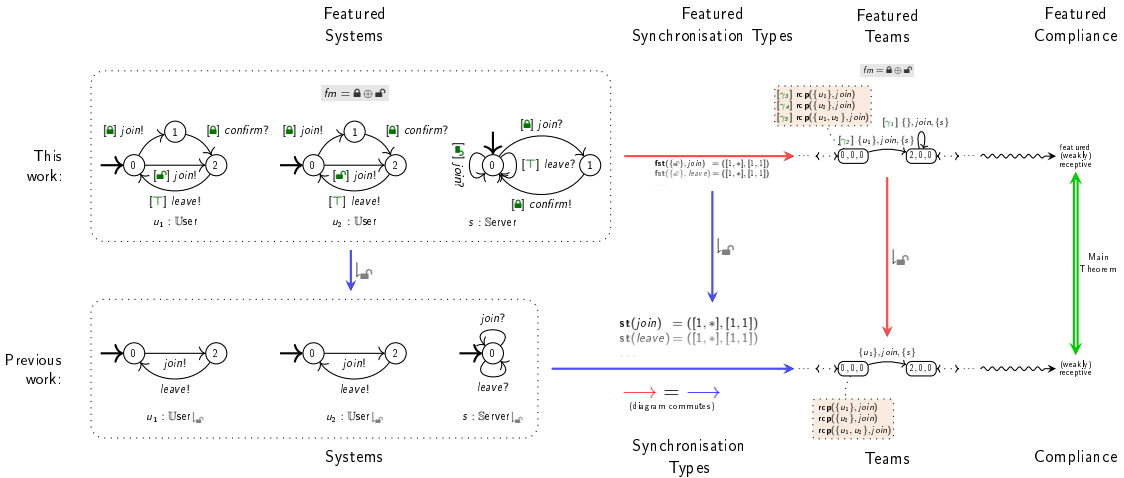
Online prototype

- Specify
- Generate*
- Visualise
- Statistics

*SAT solver to solve *fm*

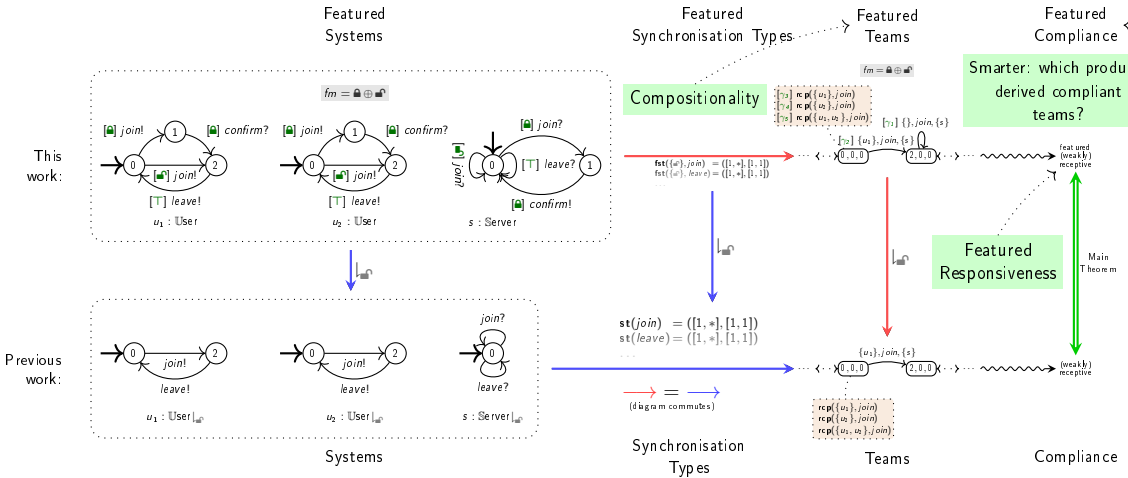


Wrapping up



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

Future work


Featured Systems
Featured Synchronisation Types
Featured Teams
Featured Compliance
This work:
Previous work:
Compositionality
Smarter: which products derived compliant teams?
Featured Responsiveness
Main Theorem
join!
leave!
confirm?
confirm!
[T] leave!
u1 : User
u2 : User
s : Server
(diagram commutes)
Systems
Synchronisation Types
Teams
Compliance
Online prototype: http://arcatools.org/feta
Extentions: e.g. check compliance

Thank you for your attention!
Questions?