Artificial Institutions supporting the regulation in Multi-Agent Systems

Maiquel de Brito

[1]UFSC-Blumenau-Brazil

WESAAC 2019 Florianópolis - Brazil

May 2019

Outline

1. Societies of agents
2. Regulative social concepts in agent societies
3. Institutions in human societies
4. Institutions in agent societies

MAS: Societies of agents

MAS are societies of agents [Shoham and Tennenholtz, 1992, Castelfranchi, 2000]

- cooperation, competition, negotiation, etc.

Computer programs in the edge between digital and social

MAS: Societies of agents - an example [Geiger, 2011]

Wikipedia Arbitration Committee:

- The wikipedia supreme court
- Solve disuptes when there is no clear community consensus

3rd annual election (2006)

- Eligibility requirement: at least 1.000 edits in the month preceeding the election
- About 40 editors meet the requirement

A unusual candidate: *AntiVandalBot*

- A computer program that reverts vandalism acts
- Reverted the deletion of its candidature

MAS: Societies of agents - an example

An essential question:

Can computadorized editors be aware about
the whole social apparatus of an arbitration comitee?

MAS: Societies of agents

It is not enough to design the agents

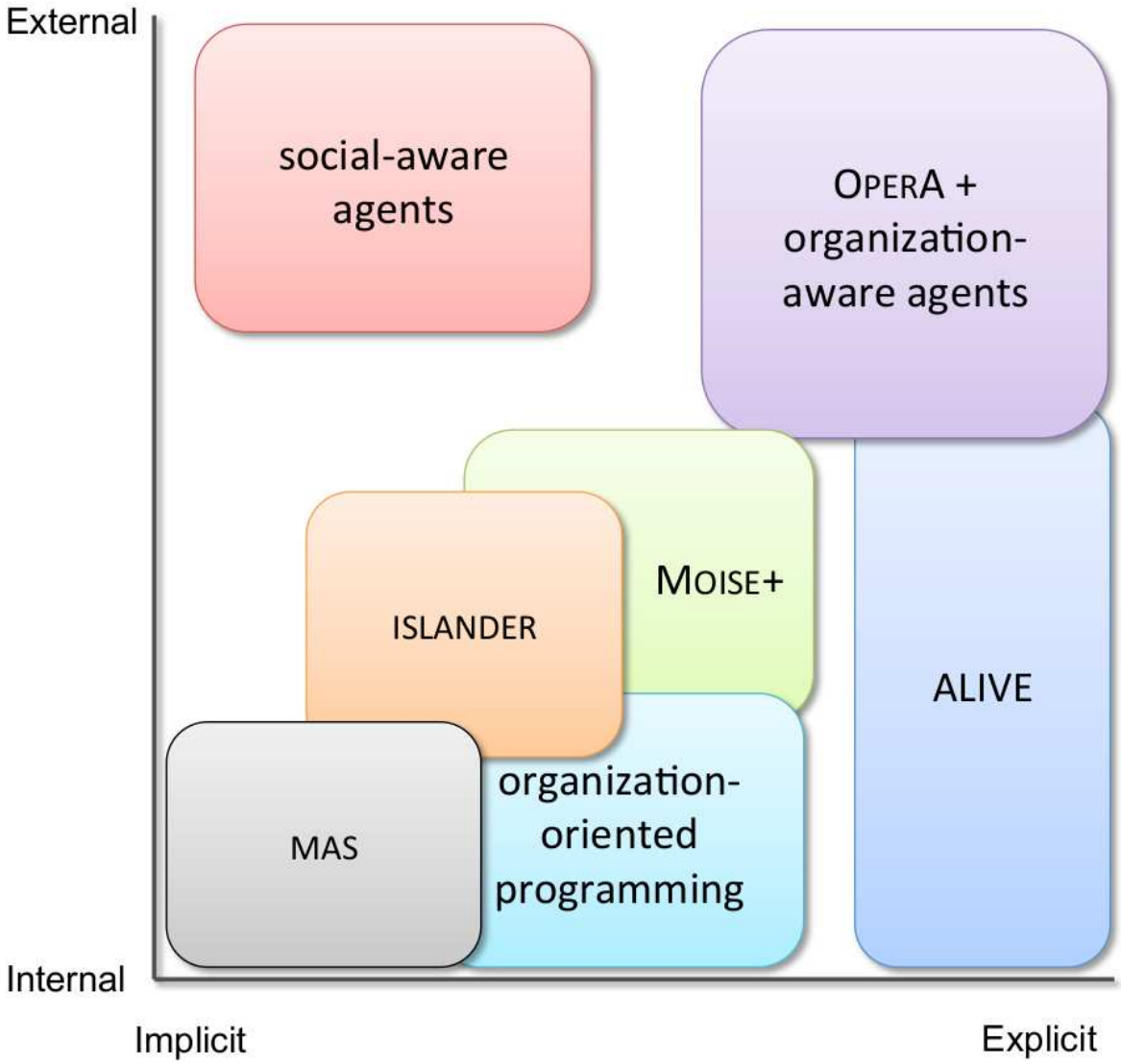
It is necessary to design the society

Social concepts in MAS

Inspiration: human societies

- Organizations
- Commitments
- Norms
- Interaction scenes
- Accountability
- Slavery
- etc.

MAS: Societies of agents



[Dignum et al. 2011]

Social abstractions in MAS

- Organizations
- Commitments
- Norms
- Interaction scenes
- etc.

Focus on regulation

agents' autonomy vs systems' expectations.

what the agents should do, when they should do, etc.

Social abstractions in MAS

- Organizations
- Commitments
- **Norms**
- Interaction scenes
- etc.

Focus on regulation

agents' autonomy vs systems' expectations.

what the agents should do, when they should do, etc.

Norms

Obligations

e.g. a PhD student is obliged to finish his thesis on time

Permissions

e.g a PhD student is permitted to finish his thesis before the final time

Prohibitions

e.g. a PhD student is prohibited to committing any plagiarism in his thesis

Several normative models, tools, etc.
Some sources: COIN series of workshops
Dagstuhl Seminar Series on Normative MAS

Regulation abstracting from environment

High level of abstraction to cover different circumstances

[Aldewereld et al. 2010]

e.g. *bidder is obliged to bid*

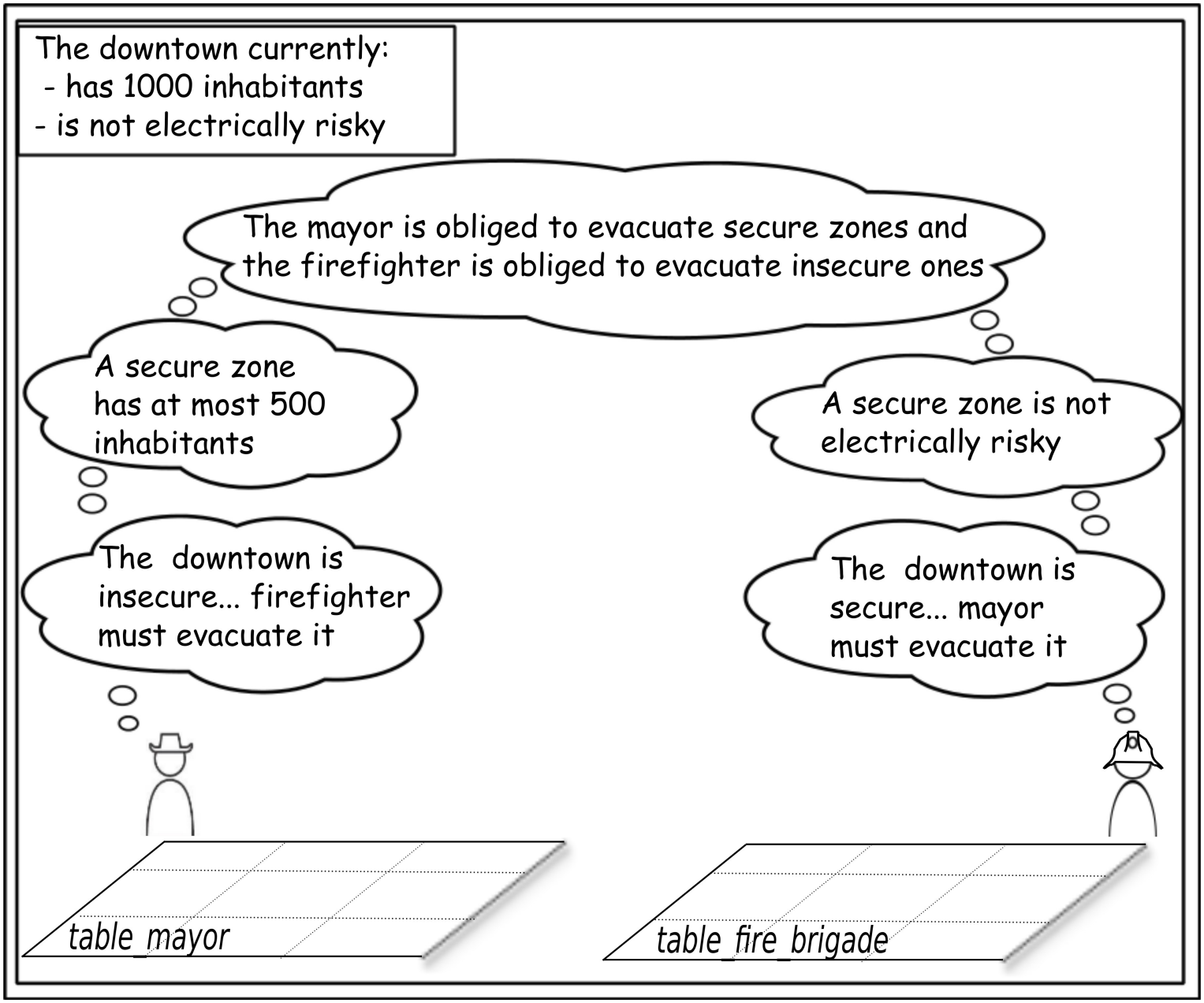
winner is obliged to pay the offer

- what is "bidder"? what is "winner"? who is obliged?
- what does it mean "to bid" and "to pay"?

Problems:

- Social abstractions represented through agentive constructs (implicit)
- Interpretation depends on the agents (internal)

The problem


The downtown currently:
- has 1000 inhabitants
- is not electrically risky
The mayor is obliged to evacuate secure zones and the firefighter is obliged to evacuate insecure ones
A secure zone has at most 500 inhabitants
A secure zone is not electrically risky
The downtown is insecure... firefighter must evacuate it
The downtown is secure... mayor must evacuate it
table_mayor
table_fire_brigade

Some questions

Problems:

- Social abstractions represented through agentive constructs (implicit)
- Interpretation depends on the agents (internal)

Questions:

- How to represent the elements referred by the regulative social abstractions?

  make them explicit

- How to connect them with the concrete environment?

  make them external

Direction: the social reality theory [Searle 1995, Searle 2009]

The social reality theory [Searle 1995, Searle 2009]

A piece of paper is a five dollar bill. Why? [Searle 1995]

"... from protons to presidents ..." [Searle 2009]

Institutional reality:

An objective portion of reality composed of abstract elements

abstract: elements are not part of the concrete world, not explainable by the basic sciences

objective: the truth of the facts is does not depend on the particular beliefs

Supports the deontology of human societies

Enabled by institutions

Institutions

Many definitions:

- Structures that constrain the individuals' behaviour

  [North 1994, Crawford e Ostrom 1995, Miller 2012]

- Not just constrain but also enable the constraining structures

  [Commons 1934, Hodgson 2006]

Institutions

Many definitions:

- Structures that constrain the individuals' behaviour

  [North 1994, Crawford e Ostrom 1995, Miller 2012]

- Not just constrain but also enable the constraining structures

  [Commons 1934, Hodgson 2006]

  - **count as**: elements from the brute reality count as elements in the institutional reality
    e.g. a piece of paper counts as money, somebody counts as president, etc.

Institutions by Searle

Based on the following elements:

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Institutions by Searle

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Functions assigned to concrete elements

not performed due to natural virtues

examples:

- piece of paper counts as five dollar bill
- individual counts as president
- raising of hands counts as bid

Institutions by Searle

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

People agree about (or at least accept)

the functions performed

by the concrete elements (that are inherent to them)

Institutions by Searle

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Express what people are expected to do or to avoid

rights, duties, obligations, authorizations, permissions, etc.

Refer to status functions

instead of to refer to concrete elements

- a PhD student is obliged to finish his thesis on time
- a PhD student is permitted to finish his thesis before the final time
- a PhD student is prohibited to committing any plagiarism in his thesis

Institutions by Searle

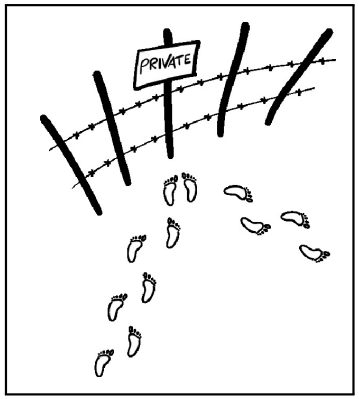
1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Status Functions + Deontic Powers:

"the glue that holds the human civilization together" [Searle 2009] - page 9

provide reasons to people act independent of

- their own desires
- physical constraints implemented in the brute reality



Institutions by Searle

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Define how status functions are constituted

connecting them to the brute reality

X counts as Y in C

e.g. A fence (X) counts as
a boundary of private property (Y)
if the property is registered to some owner (C)

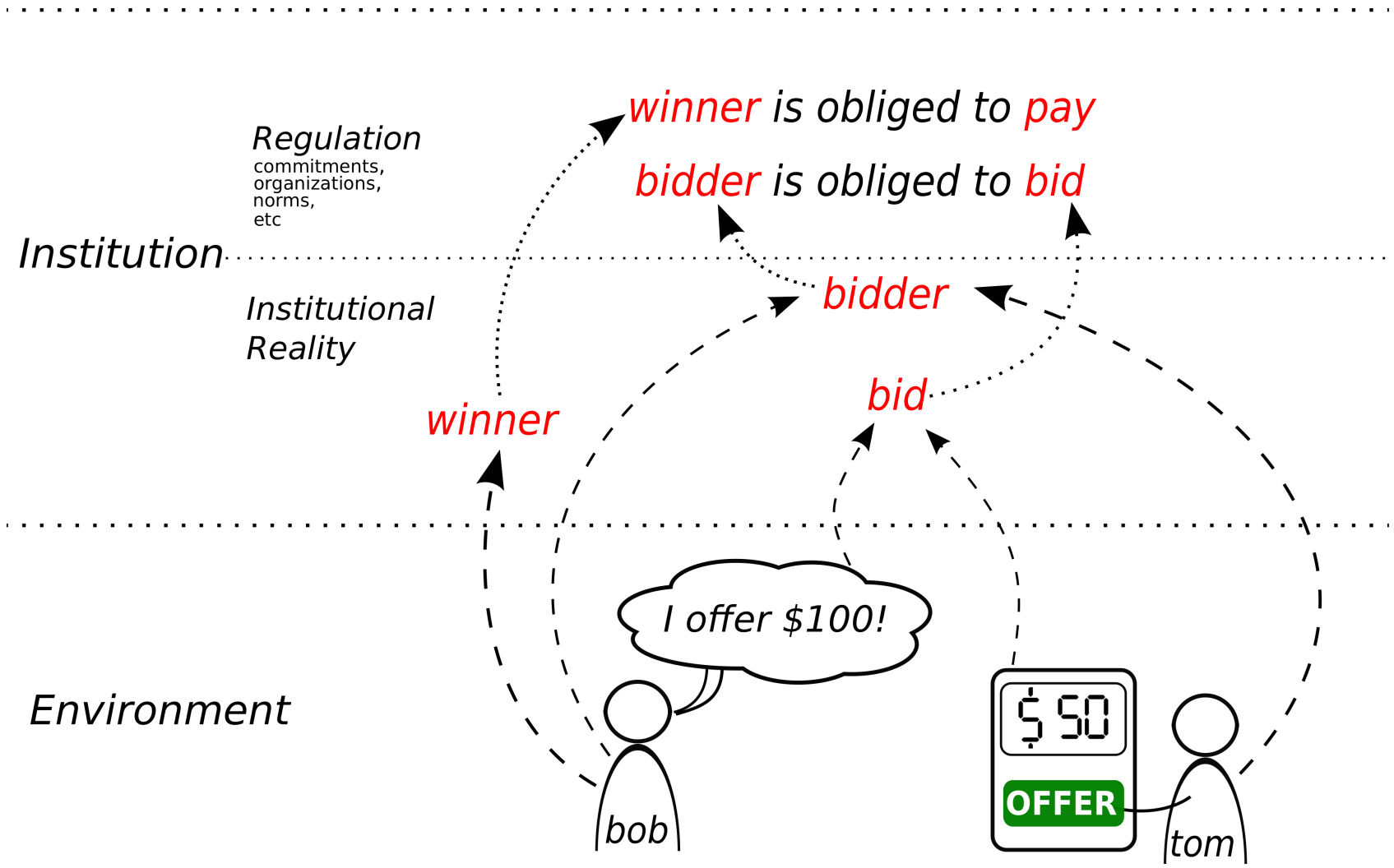
Institutions by Searle

"...institutions are systems of constitutive rules and the constitutive rules create the possibility of institutional facts" (SEARLE, 2009, page 10)

Constituted status functions

- the building block of institutions
- constituted thanks to constitutive rules
- deontic powers are attached to
- required to regulation: individuals behave as expected when status functions are constituted

Institutions in Agent Societies



Situated Artificial Institution (SAI)

Institution: Regulation + Institutional reality

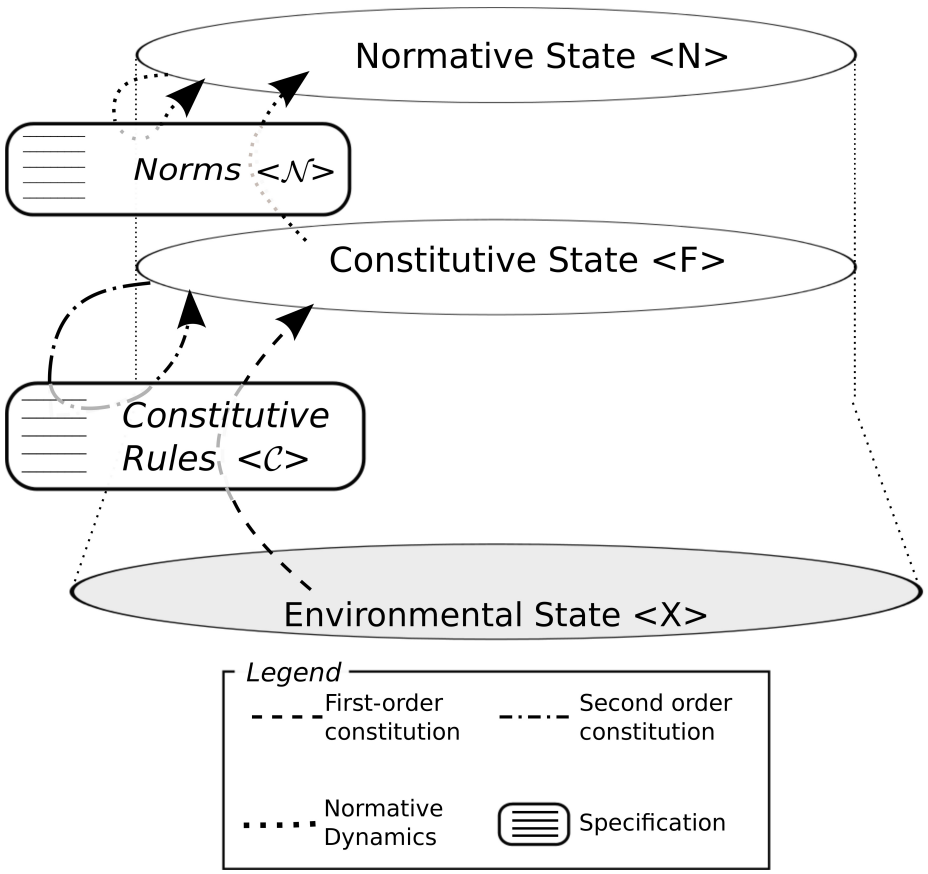
Assumption: regulation provided by existing normative models

existing works are enough

What is needed?

1. A model of institutional reality
2. Coupling existing norms in the institutional reality

Situated Artificial Institution (SAI)



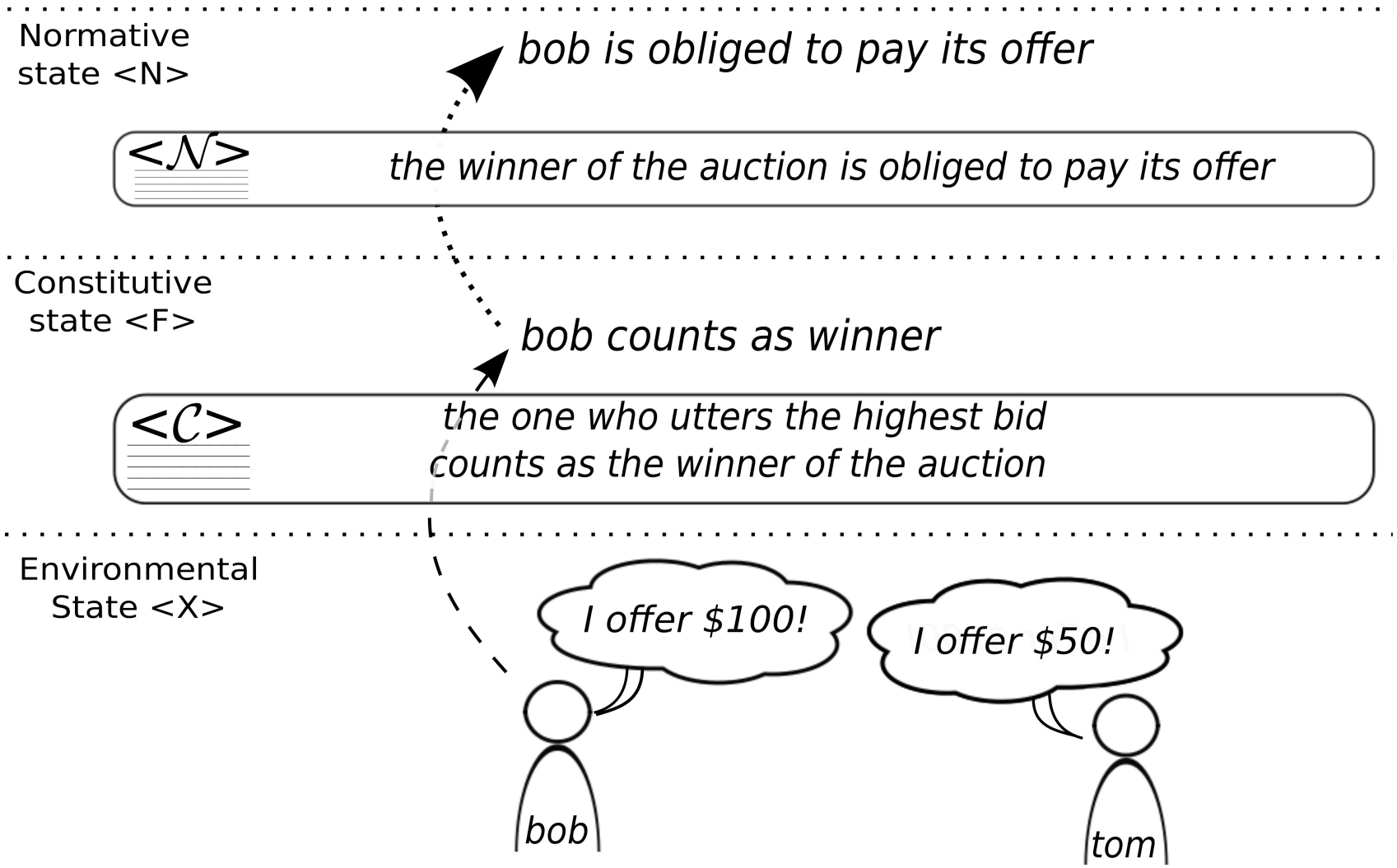
Specification

- Norms ($\mathcal{N}$):
  - define what agents must do/avoid
  - composed of status functions
- Status functions ($\mathcal{F}$): functions assignable to agents, events and states
- Constitutive rules ($\mathcal{C}$): constitute the *status functions*

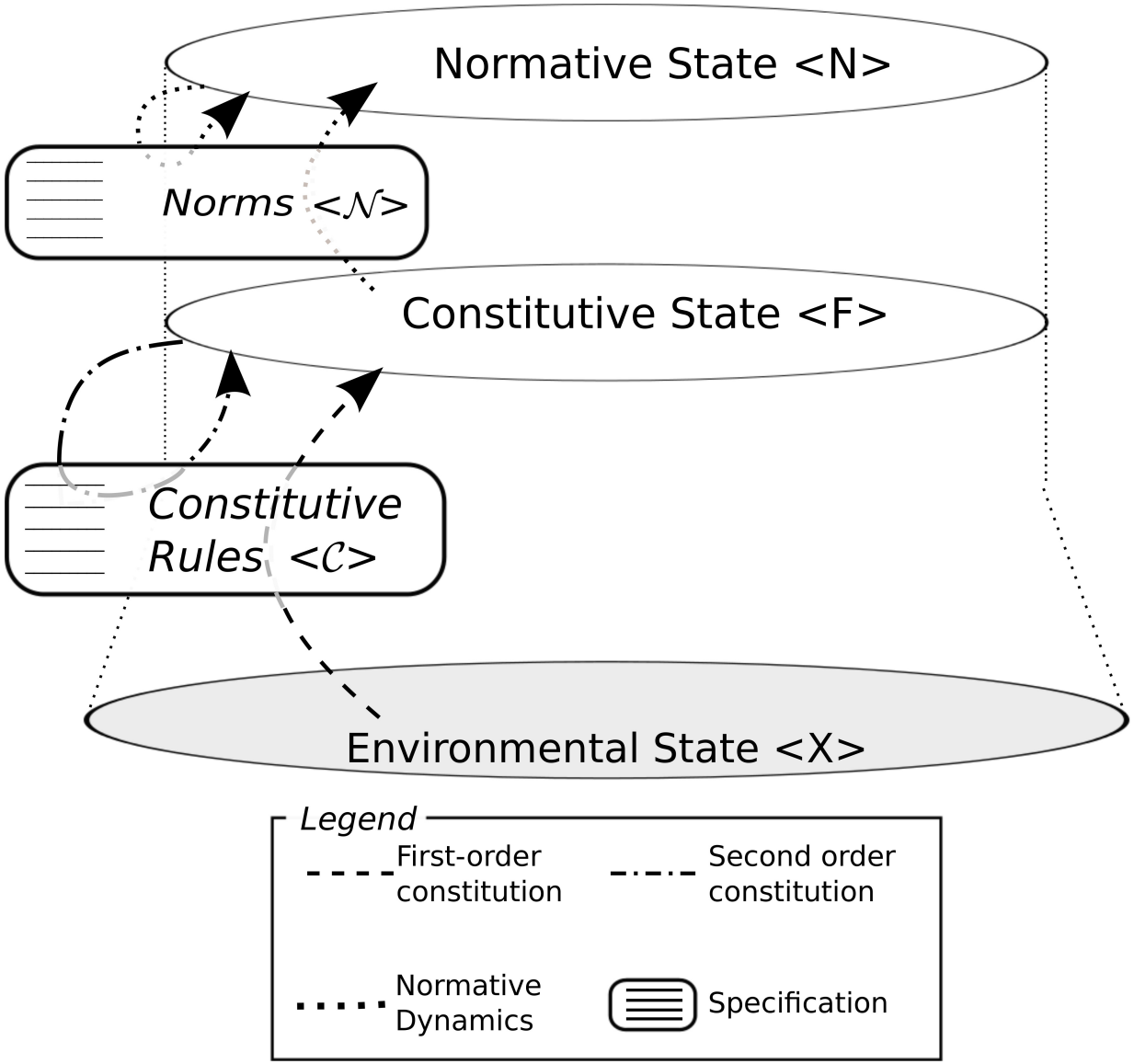
Dynamics

- Normative state (N): expected agents' behaviour
- Constitutive state (F): institutional reality

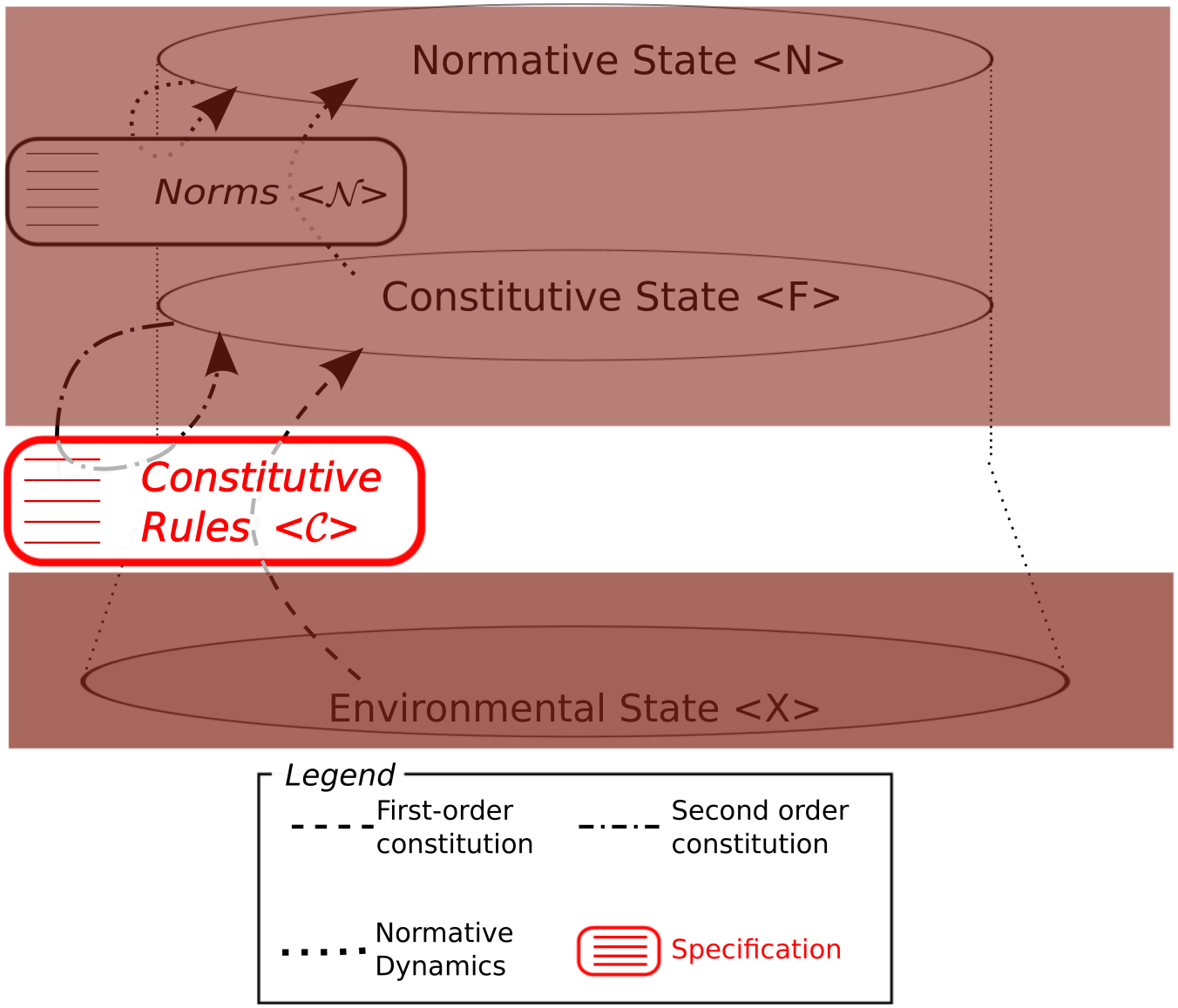
Situated Artificial Institution (SAI)



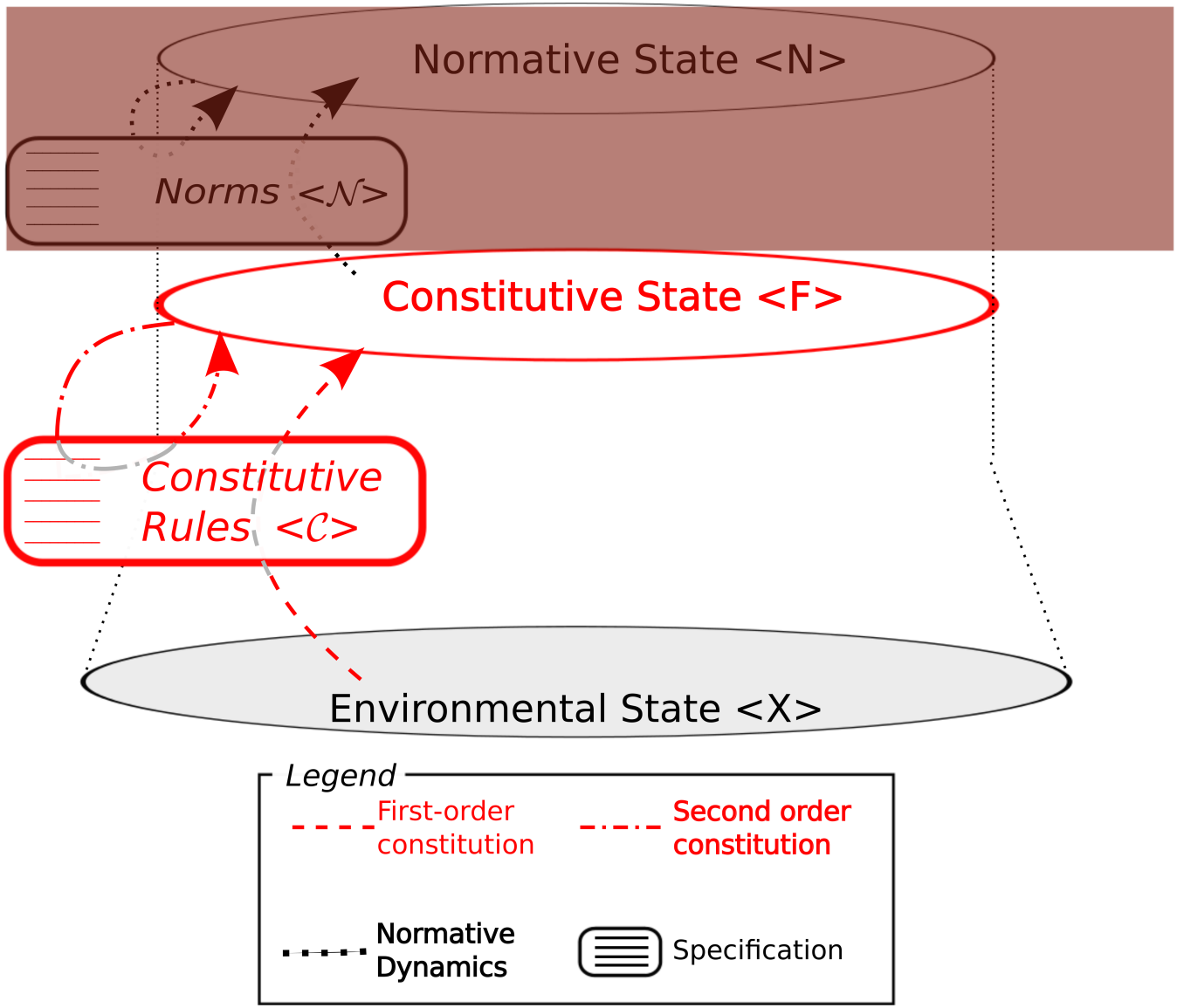
Institutional Reality in SAI



Institutional Reality in SAI



Institutional Reality in SAI



Institutional Reality in SAI - Constitutive Specification

Status functions: status assigned to agents, events, and states
that impose functions to them

Status functions in SAI: $\mathcal{F} = \mathcal{A}_{\mathcal{F}} \cup \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$ s.t.

- $\mathcal{A}_{\mathcal{F}}$: status functions assignable to **agents**, i.e. agent-status functions
  e.g.: bob counts as winner

- $\mathcal{E}_{\mathcal{F}}$: status functions assignable to **events**, i.e. event-status functions
  e.g.: tell("I offer $100") counts as bid

- $\mathcal{S}_{\mathcal{F}}$: status functions
  - assignable to **states**, i.e. state-status functions
    e.g.: "twenty agents in the system" counts as quorum for an auction
  - involved in freestanding assignment
    e.g. assigned to conditions that do not involve environmental states

Institutional Reality in SAI - Constitutive Specification

Constitutive rules: specify the constitution of status functions

Constitutive rules in SAI: $c \in \mathcal{C} = \langle x, y, t, m \rangle$ s.t.

- x has the *status function* y when t occurs, while m holds
- t and m can be omitted

Example:

$$\langle Agent, auctioneer, (propose(auction), Agent), \neg auction\_finished \rangle$$

$\downarrow$ expressed as $\downarrow$

```
Agent count-as auctioneer
      when (propose(auction),Agent)
      while not auction_finished.
```

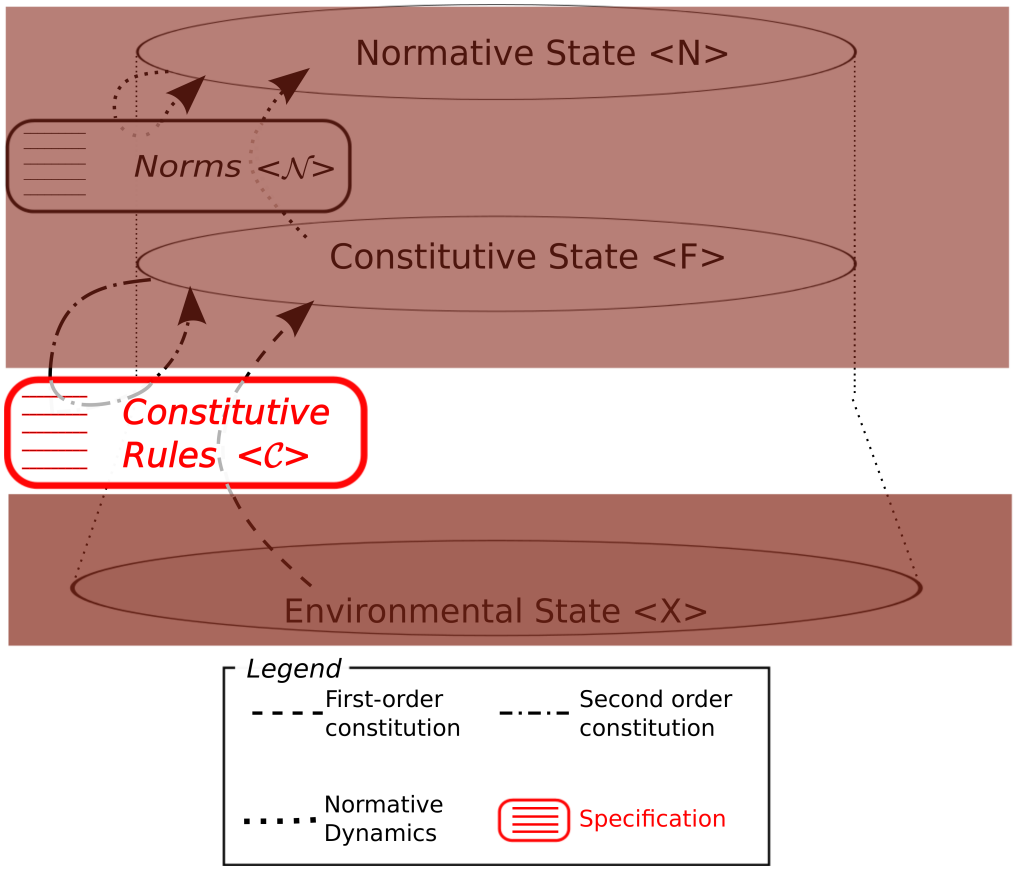
Institutional Reality in SAI - Constitutive dynamics

Interpretation of constitutive rules

- Produces constitution of status functions

Constitutive state (F)

- Set of constituted status functions
- the SAI representation of institutional reality



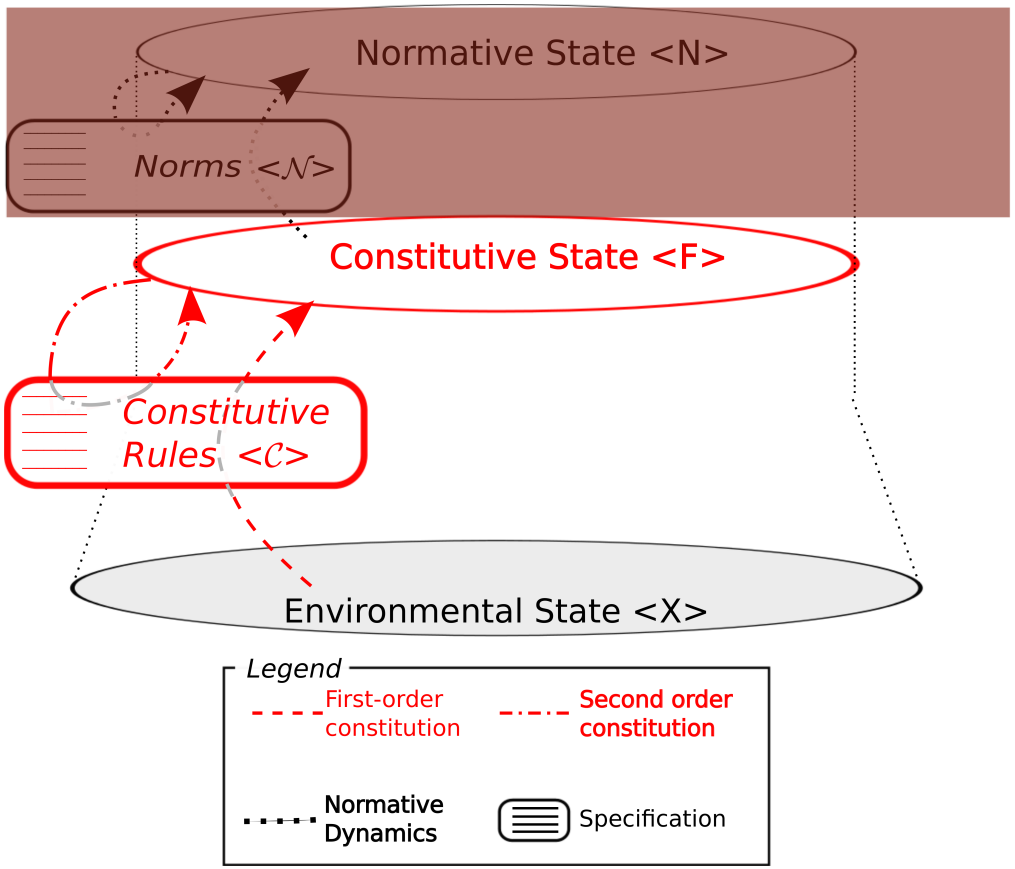
Institutional Reality in SAI - Constitutive dynamics

Interpretation of constitutive rules

- Produces constitution of status functions

Constitutive state (F)

- Set of constituted status functions
- the SAI representation of institutional reality



Constitutive dynamics - 1st and 2nd order constitution

- First-order constitution

  Status functions assigned to environmental elements

- Second-order constitution

  Status functions assignments lead to new status functions assignments

Constitutive dynamics - 1st and 2nd order constitution

- **Environmental state :** $X = A_X \cup E_X \cup S_X$

  where A_X =agents acting; E_X=events occurring; S_X = states holding

- **Constitutive state :** $F = A_F \cup E_F \cup S_F$

  where A_F =agent-SFAs ; E_X=event-SFAs; S_X = state-SFAs

1st order constitution of agent-status functions

$$f\text{-}const_a(\mathcal{F}, \mathcal{C}, X^0 \cdots X^z, F^0 \cdots F^z, i) = \{\langle x, y\rangle | x \in A_X^i \wedge y \in \mathcal{A}_{\mathcal{F}} \wedge \exists c \in \mathcal{C}\ \exists \theta\ \exists s \in \mathbb{N}\ \forall k \in [s, i] : (E_X^s \cup\ E_F^s \models t\theta) \wedge (X^k \cup F^k \models m\theta) \wedge x'\theta = x\}$$

$$s.t.\ c = \langle x', y, t, m\rangle$$

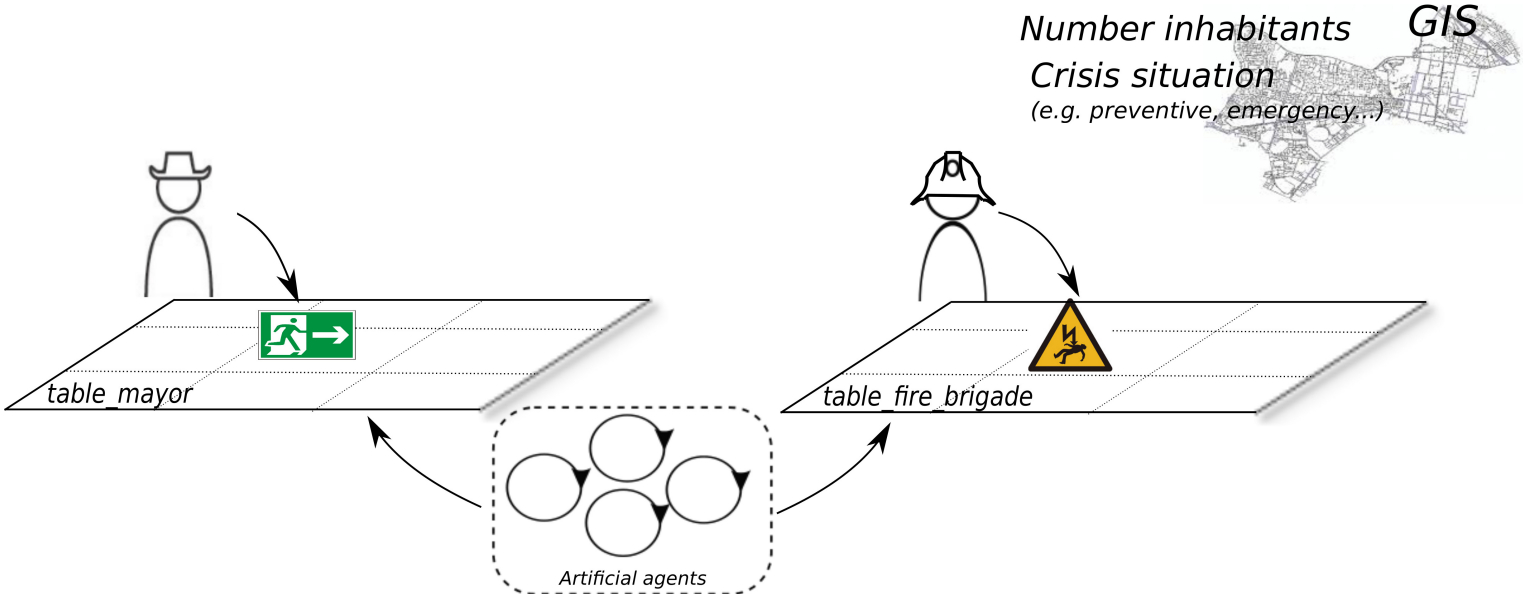
2nd order constitution of agent-status functions

$$s\text{-}const_a(\mathcal{F}, \mathcal{C}, X^0 \cdots X^z, F^0 \cdots F^z, i) = \{\langle x, y\rangle | x \in A_X^i \wedge y \in \mathcal{A}_{\mathcal{F}} \wedge \exists c \in \mathcal{C}\ \exists \theta\ \exists s \in \mathbb{N}\ \forall k \in [s, i] : (E_X^s \cup\ E_F^s \models t\theta) \wedge (X^k \cup F^k \models m\theta) \wedge x'\theta \in \mathcal{A}_{\mathcal{F}} \wedge \langle x, x'\theta\rangle \in A_F^i\}$$

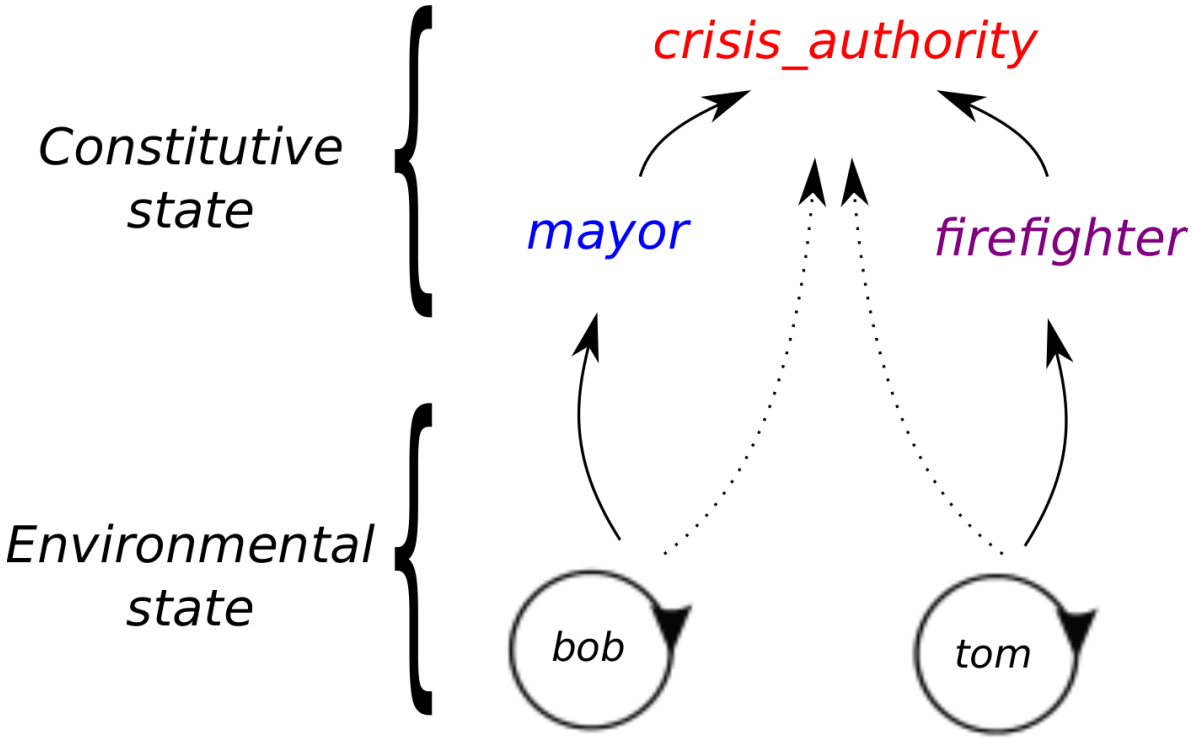
$$s.t.\ c = \langle x', y, t, m\rangle$$

Constitutive dynamics - example scenario

crisis management



Constitutive dynamics - 1st and 2nd order constitution

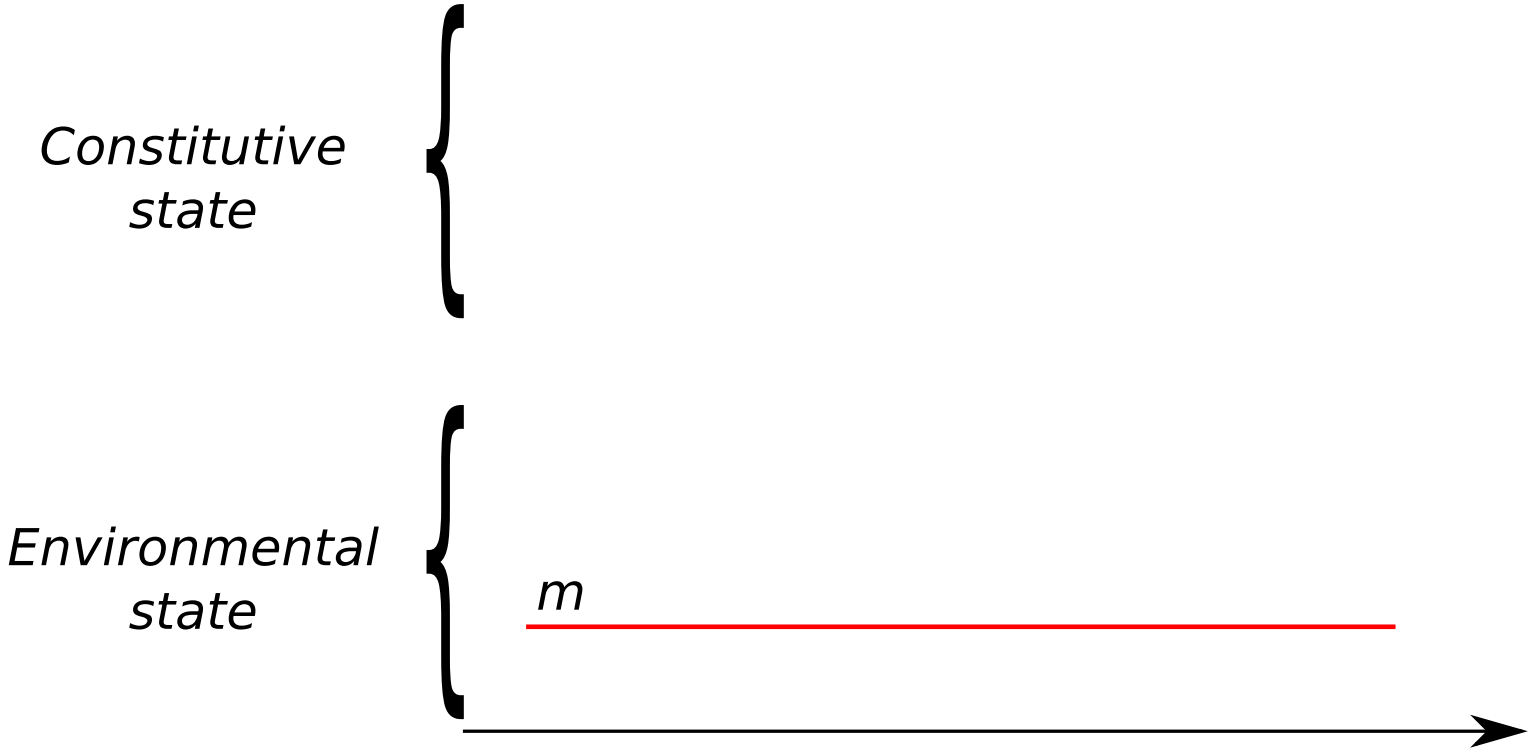


1st order { bob count as mayor; tom count as firefighter }

2nd order { mayor count-as crisis_authority; firefighter count-as crisis_authority }

Constitutive dynamics - 1st order constitution

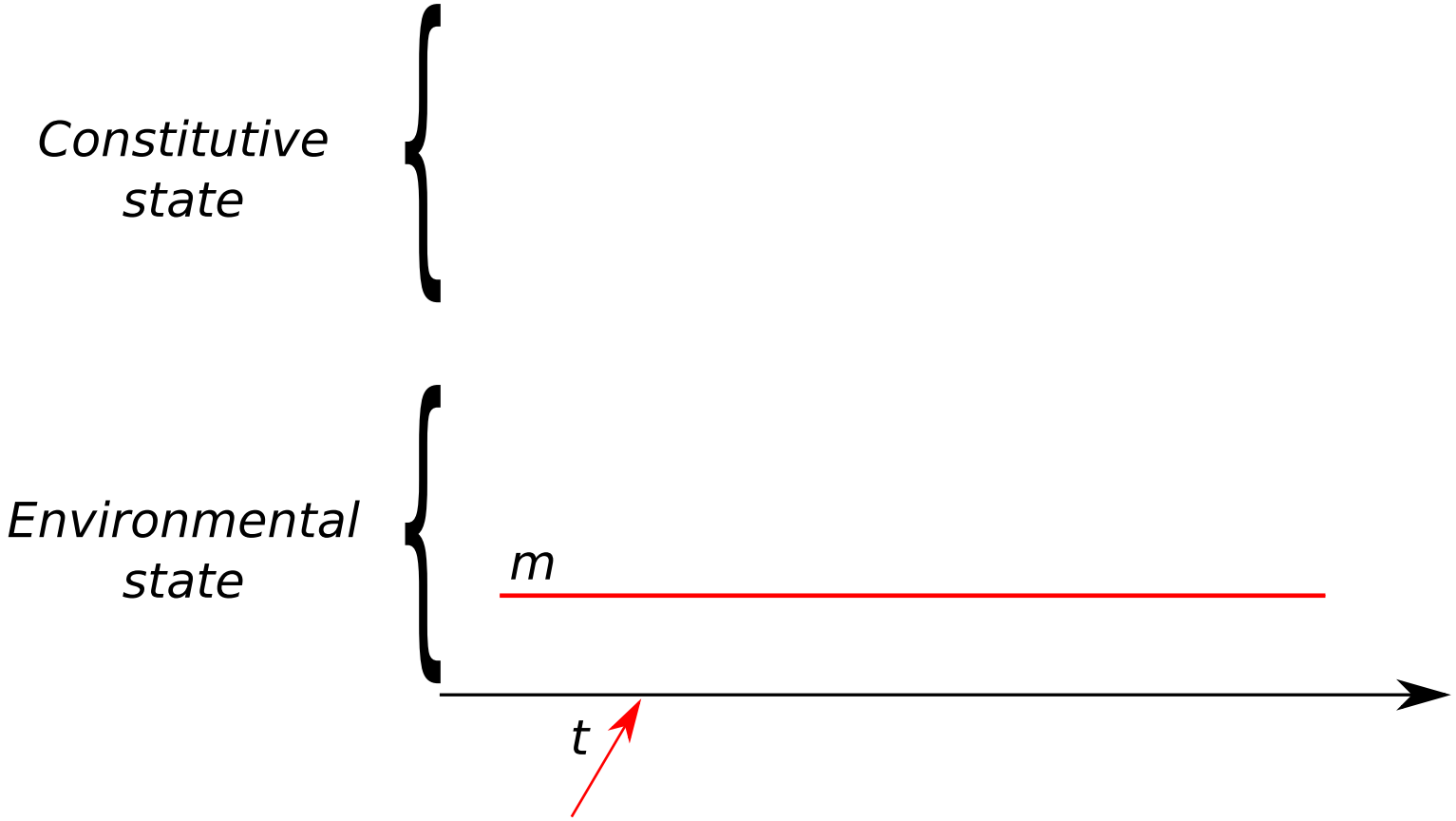
Agent- and state-status function



x count-as y when t while m

Constitutive dynamics - 1st order constitution

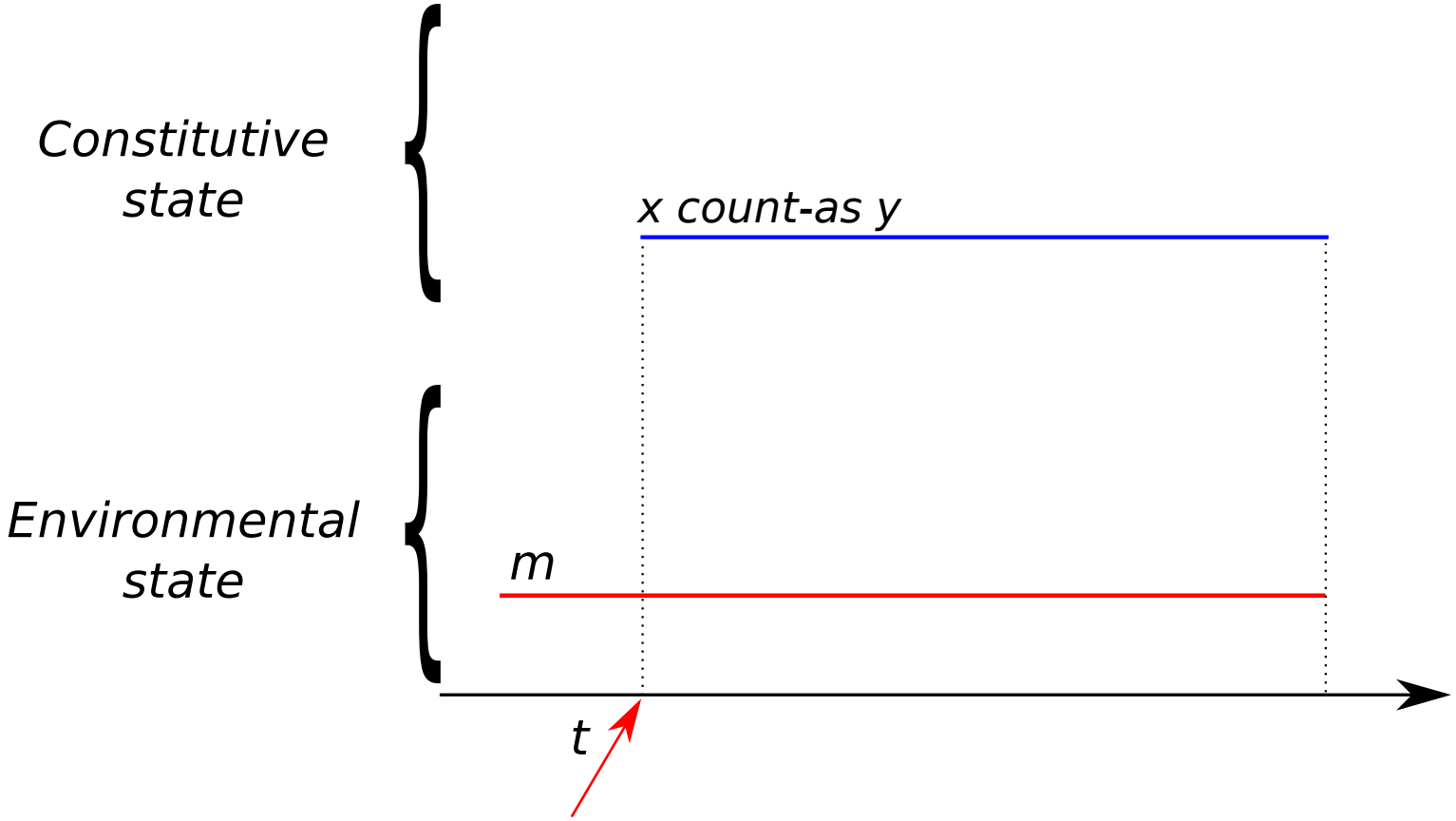
Agent- and state-status function



x count-as y when t while m

Constitutive dynamics - 1st order constitution

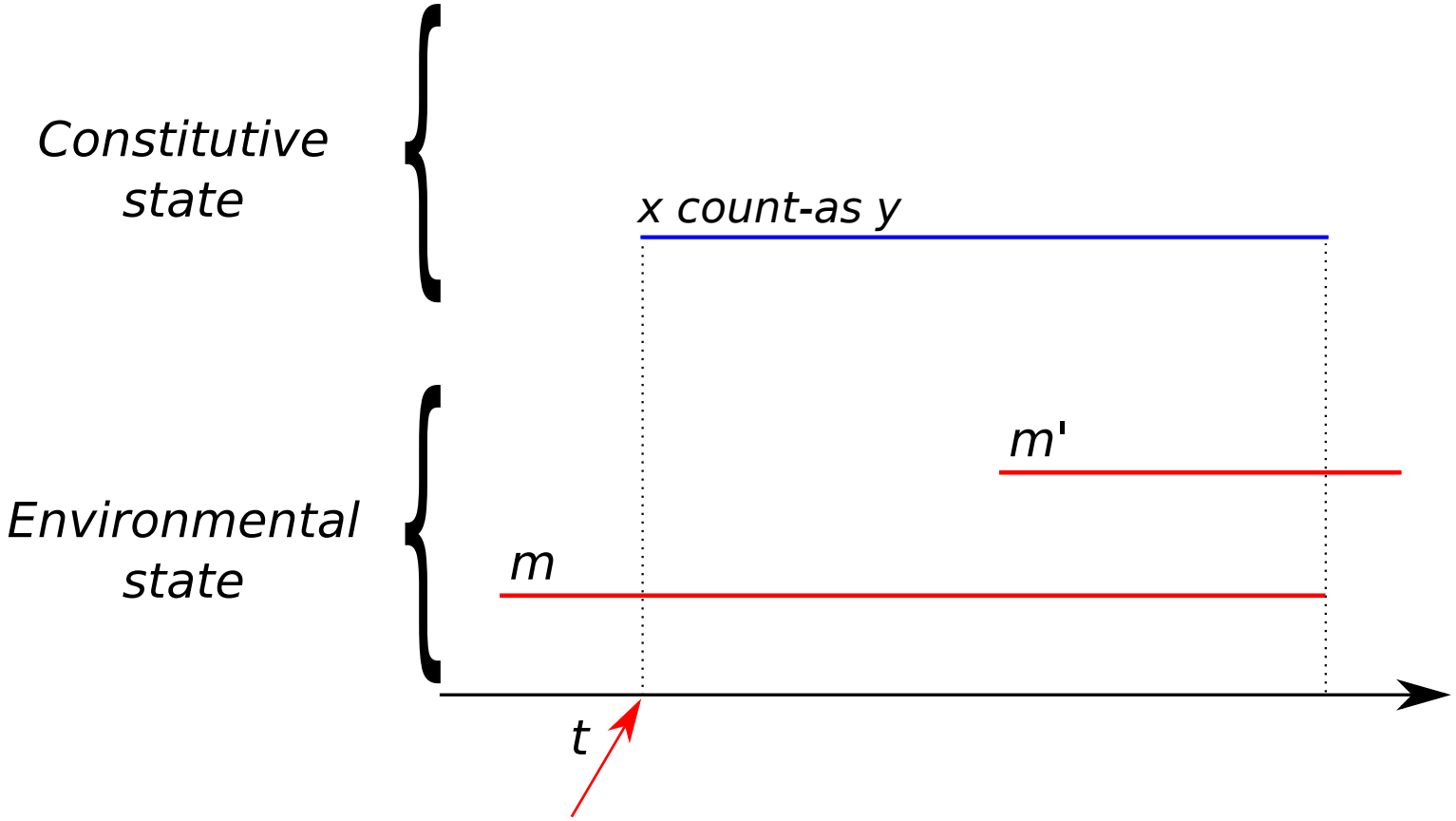
Agent- and state-status function



x count-as y when t while m

Constitutive dynamics - 2nd order constitution

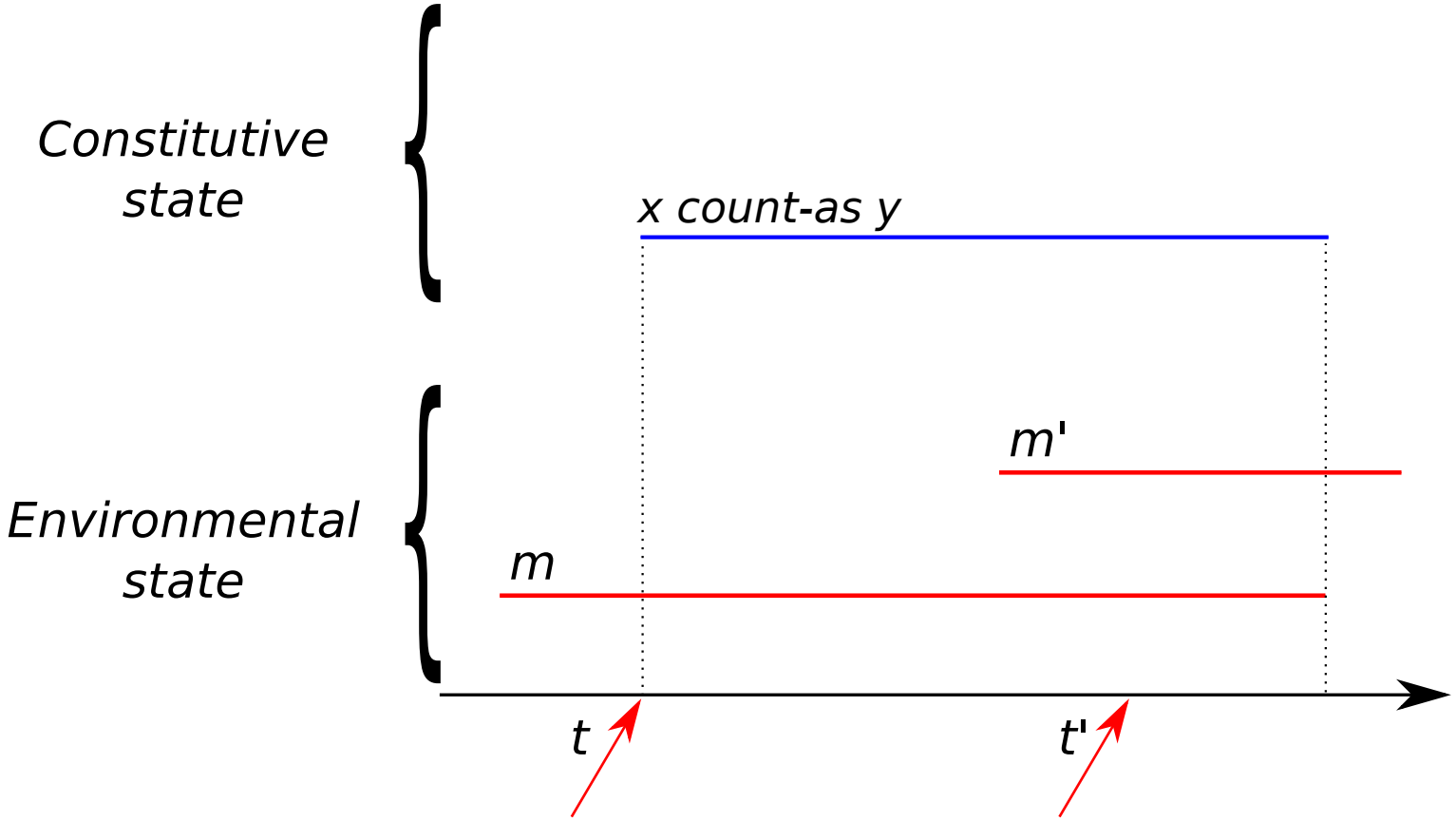
Agent- and state-status function



x count-as y when t while m
y count-as y' when t' while m'

Constitutive dynamics - 2nd order constitution

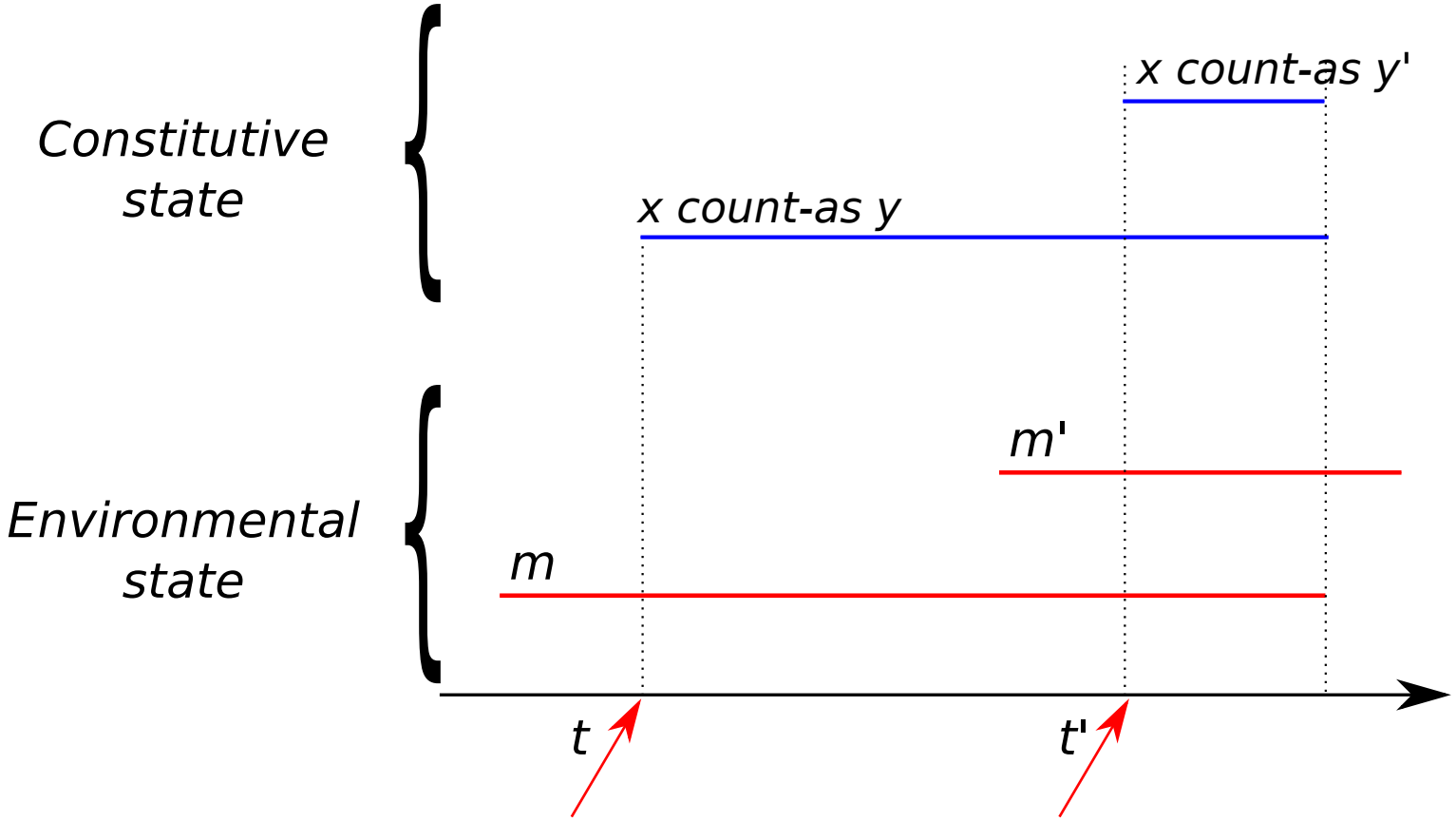
Agent- and state-status function



x count-as y when t while m
y count-as y' when t' while m'

Constitutive dynamics - 2nd order constitution

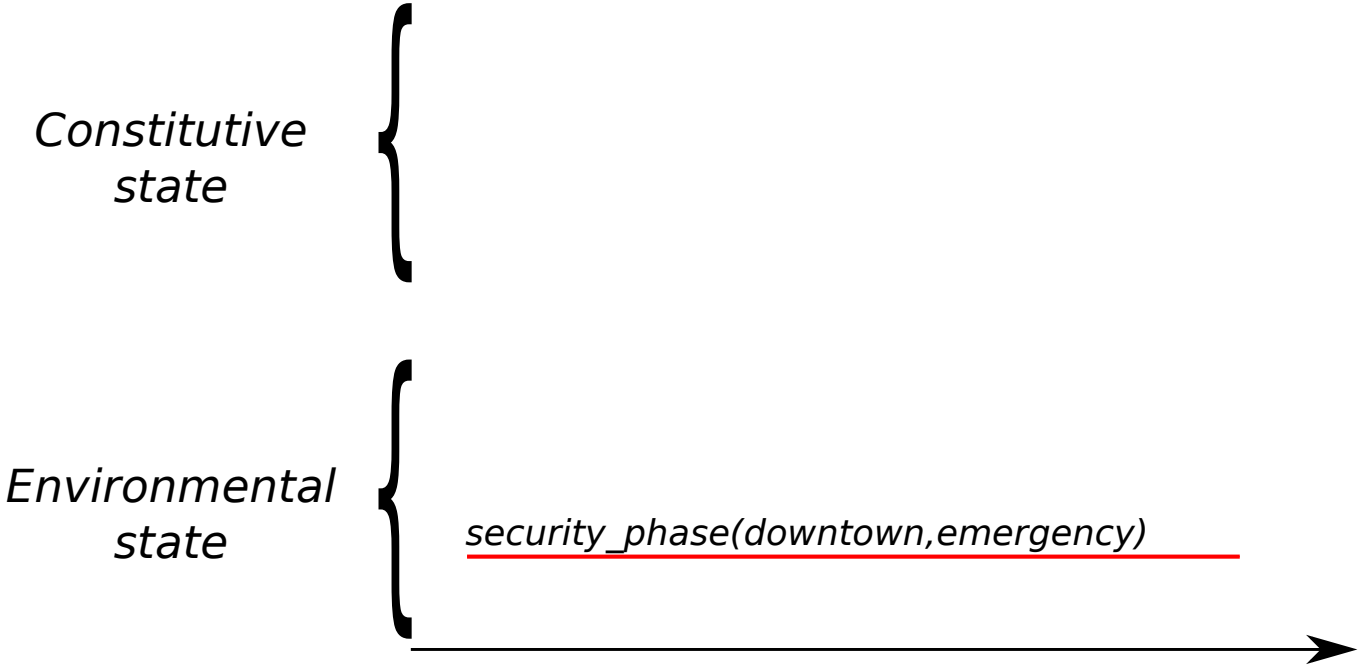
Agent- and state-status function



x count-as y when t while m

y count-as y' when t' while m'

Constitutive dynamics - Example



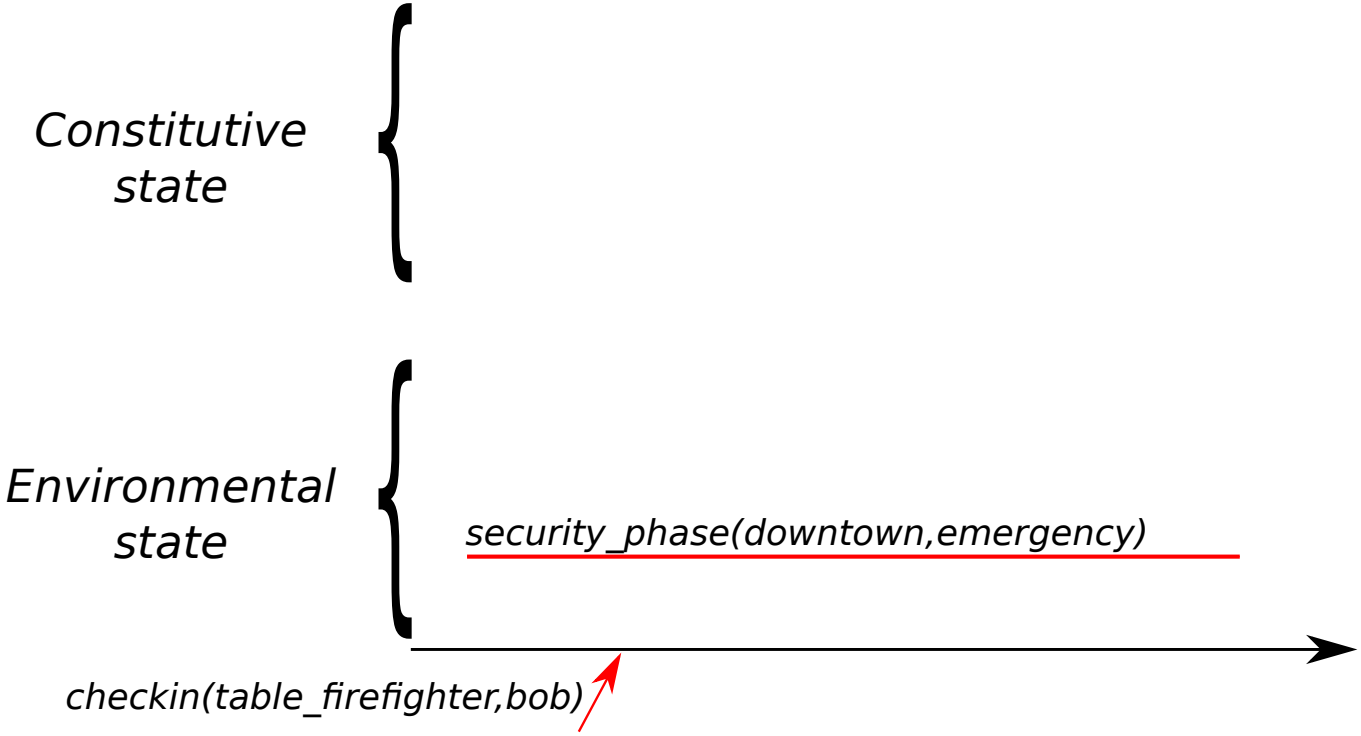
1st order

```
Agent count-as firefighter
  when checkin(table_firefighter,Agent)
  while security_phase(downtown,emergency).
```

2nd order

```
firefighter
  count-as crisis_authority
  when command_evacuation(_).
```

Constitutive dynamics - Example



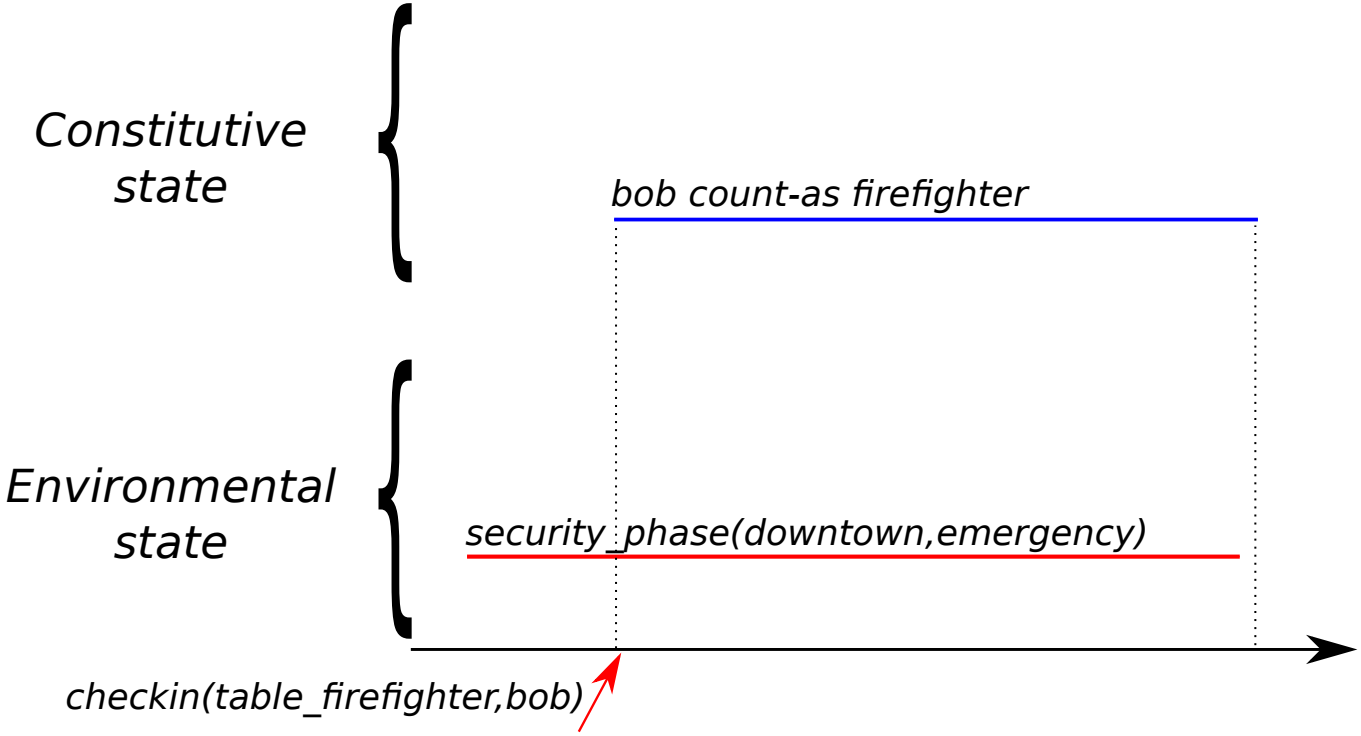
1st order

```
Agent count-as firefighter
  when checkin(table_firefighter,Agent)
  while security_phase(downtown,emergency).
```

2nd order

```
firefighter
  count-as crisis_authority
  when command_evacuation(_).
```

Constitutive dynamics - Example



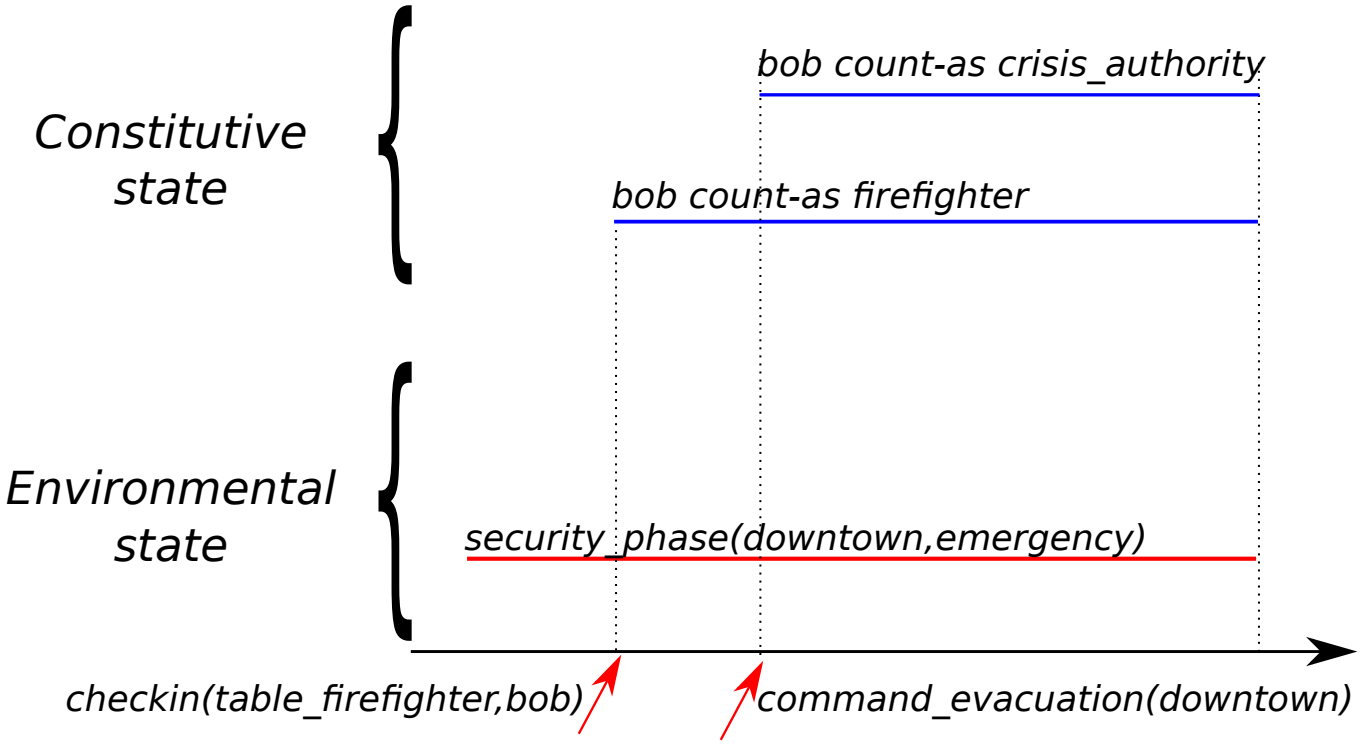
1st order

```
Agent count-as firefighter
  when checkin(table_firefighter,Agent)
  while security_phase(downtown,emergency).
```

2nd order

```
firefighter
  count-as crisis_authority
  when command_evacuation(_).
```

Constitutive dynamics - Example



1st order

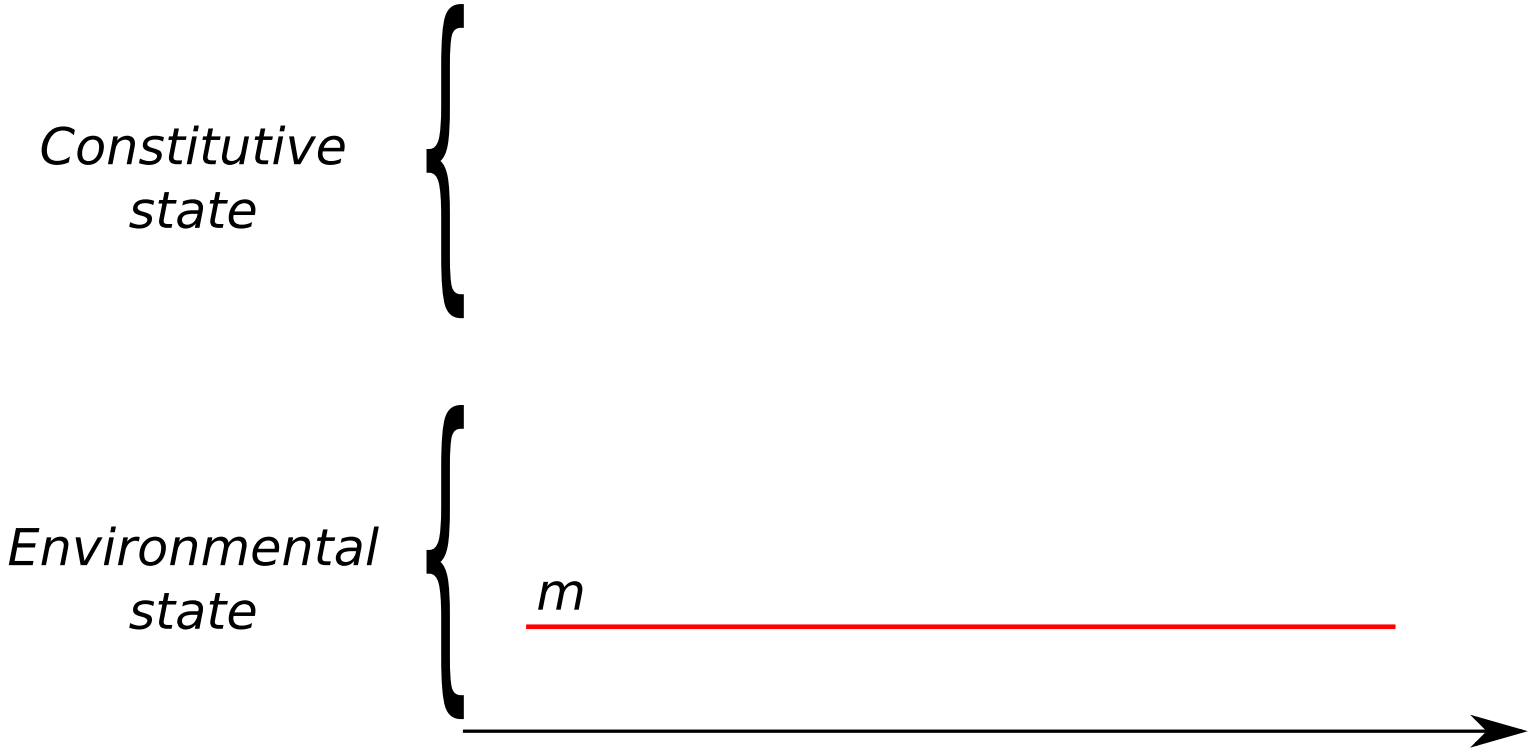
```
Agent count-as firefighter
  when checkin(table_firefighter,Agent)
  while security_phase(downtown,emergency).
```

2nd order

```
firefighter
  count-as crisis_authority
  when command_evacuation(_).
```

Constitutive dynamics - 1st order constitution

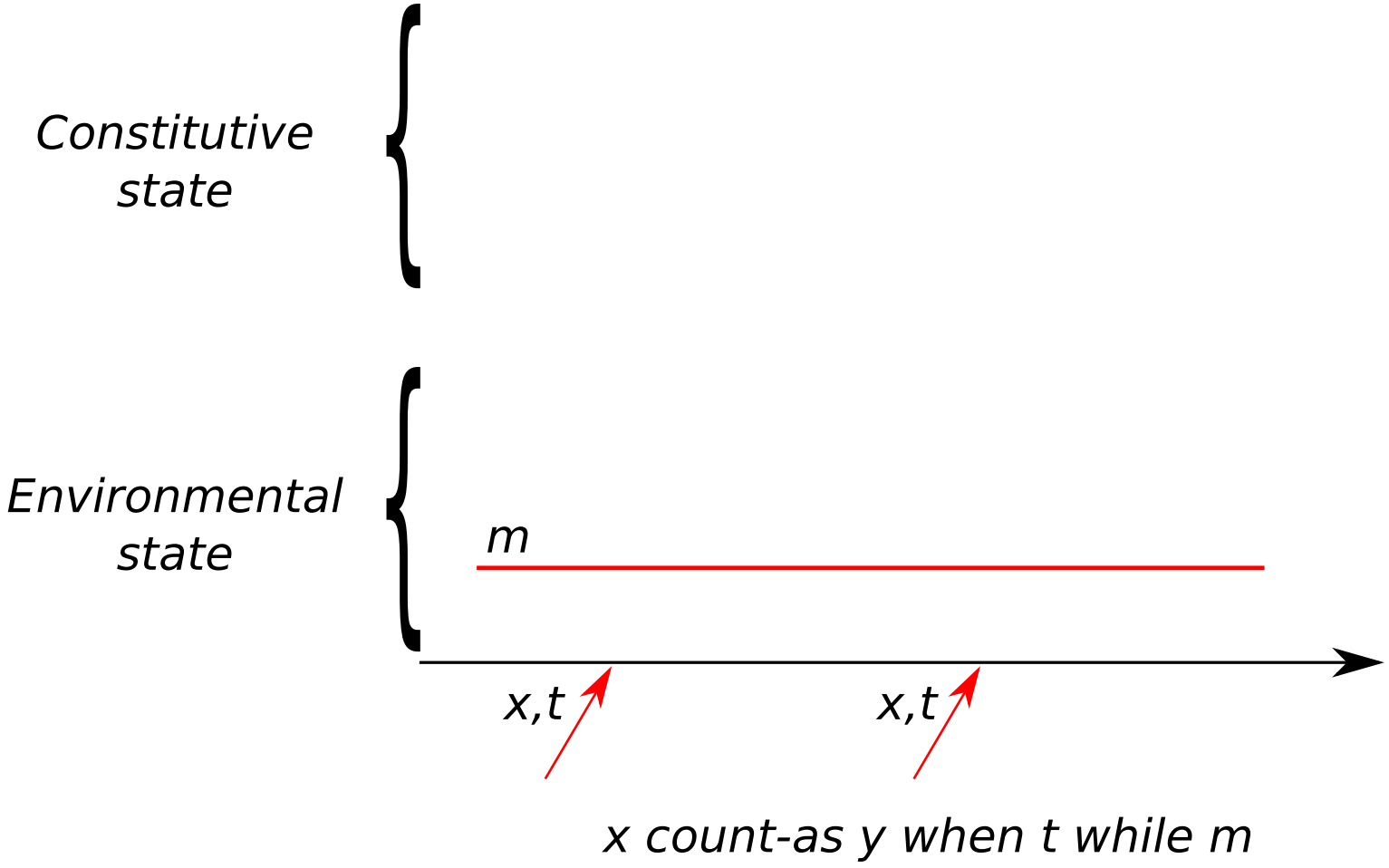
Event-status function



x count-as y when t while m

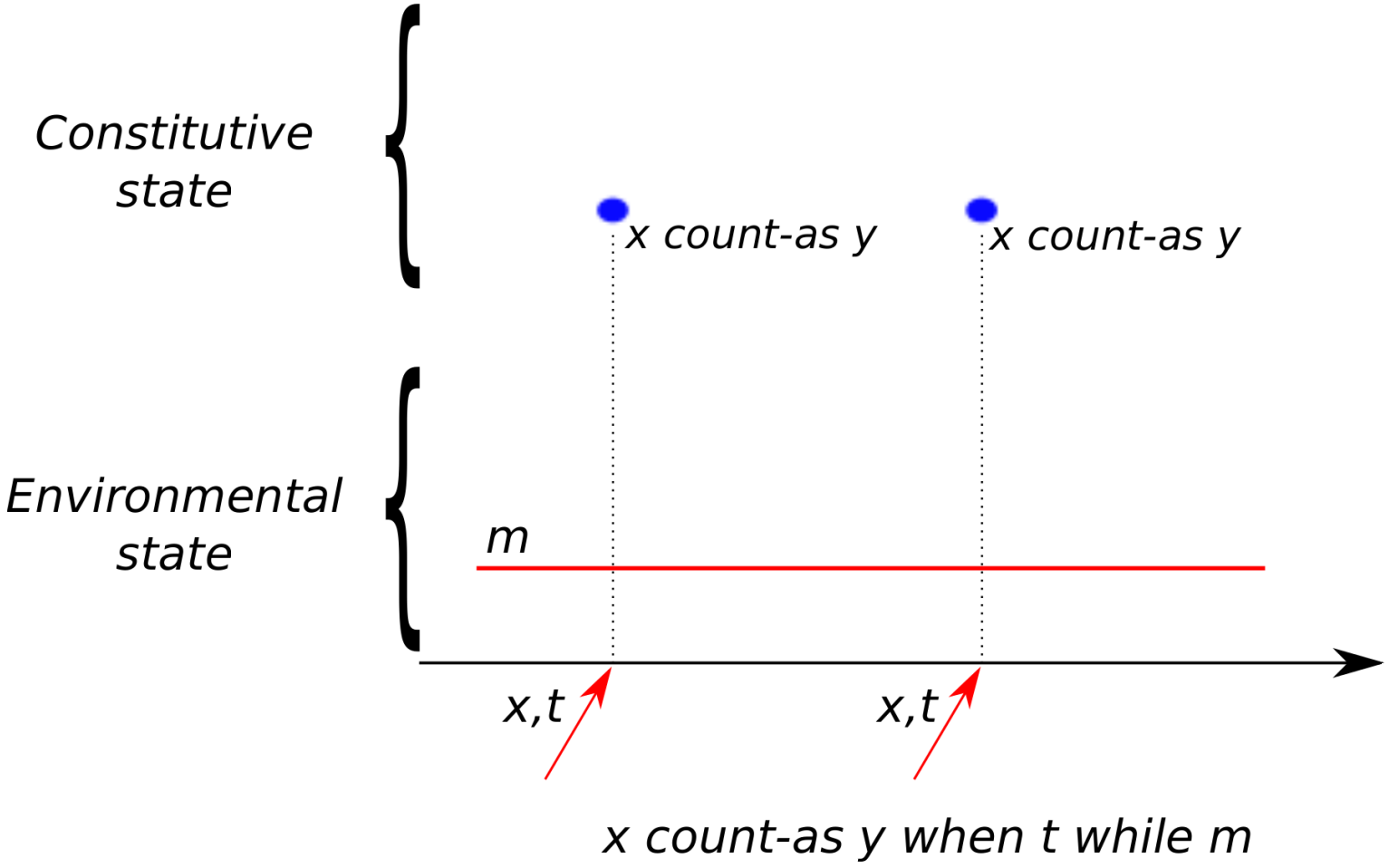
Constitutive dynamics - 1st order constitution

Event-status function



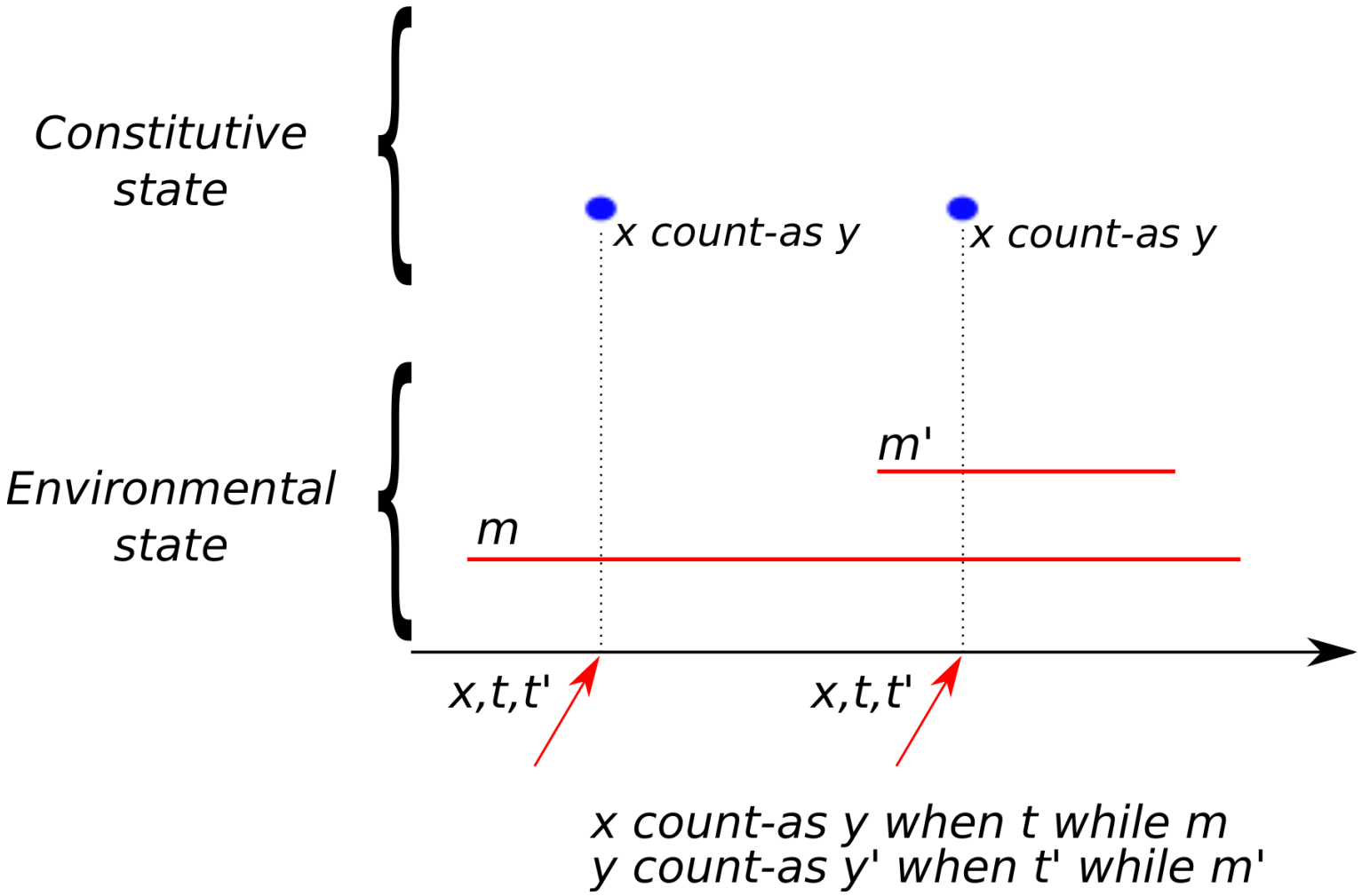
Constitutive dynamics - 1st order constitution

Event-status function



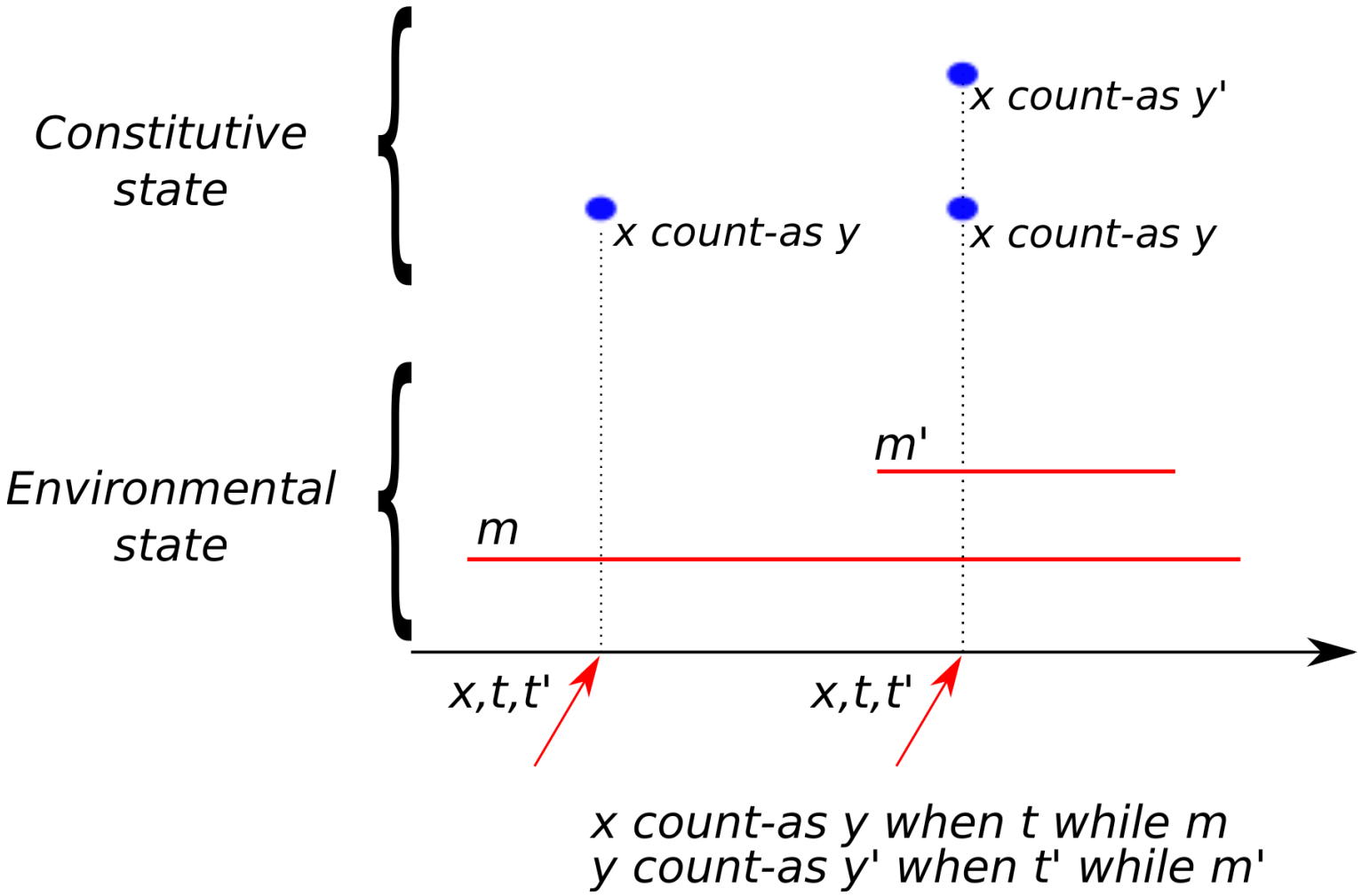
Constitutive dynamics - 2nd order constitution

Event-status function

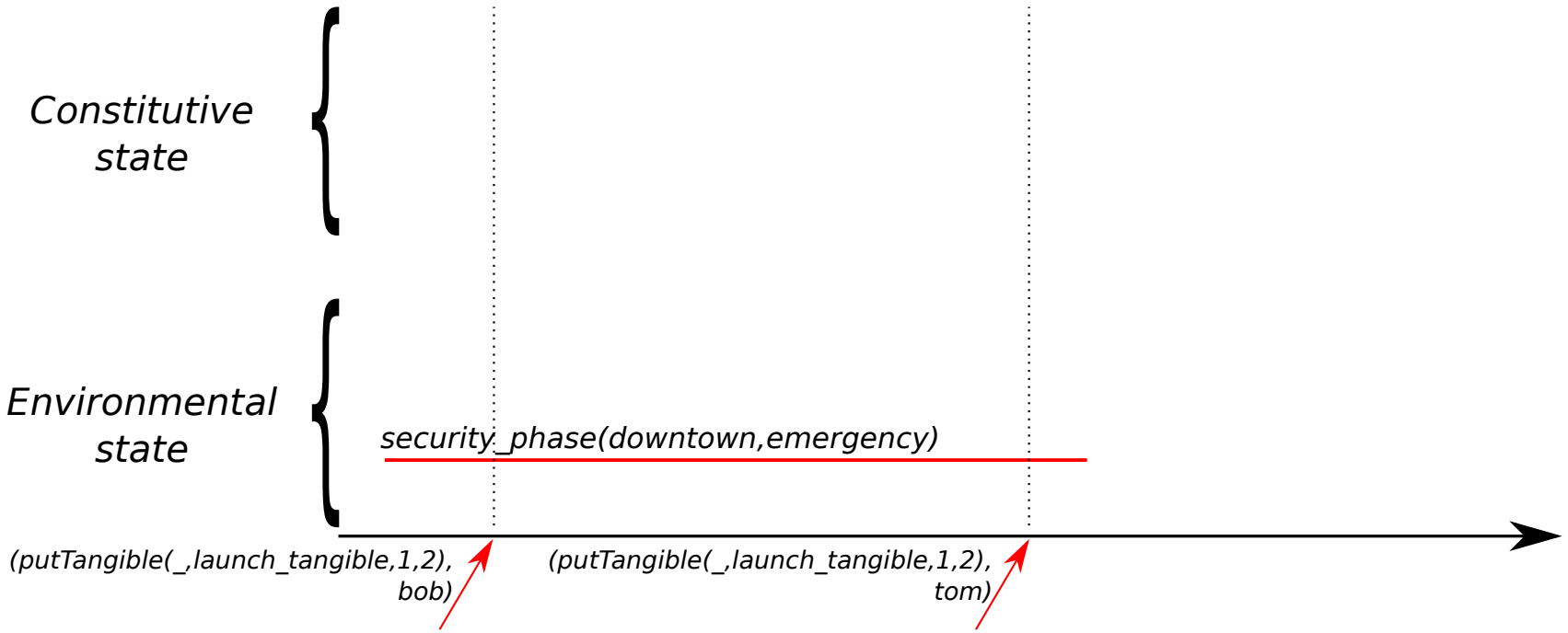


Constitutive dynamics - 2nd order constitution

Event-status function



Constitutive dynamics - example



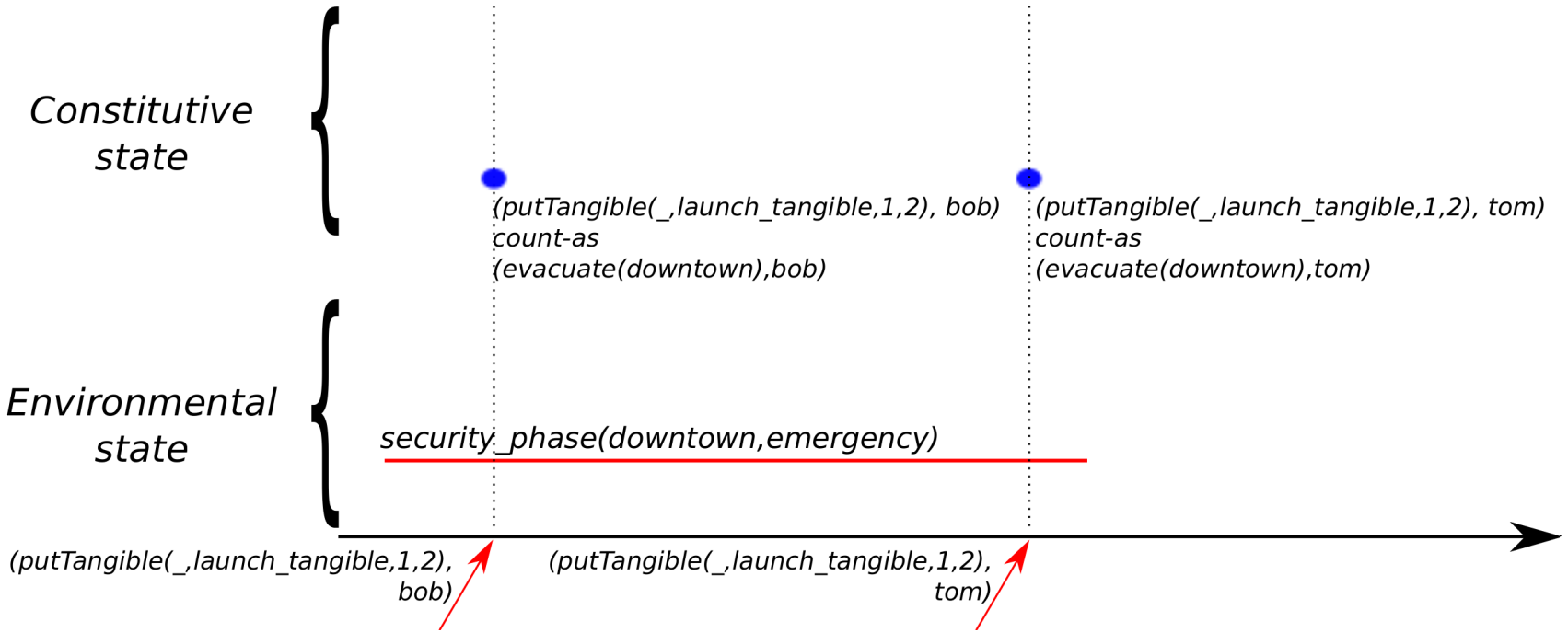
1st order

```
putTangible(_,launch_tangible,1,2)
                    [sai__agent(_)]
  count-as evacuate(downtown)
  while security_phase(downtown,emergency).
```

2nd order

```
evacuate(downtown)
  count-as security_procedure.
```

Constitutive dynamics - example



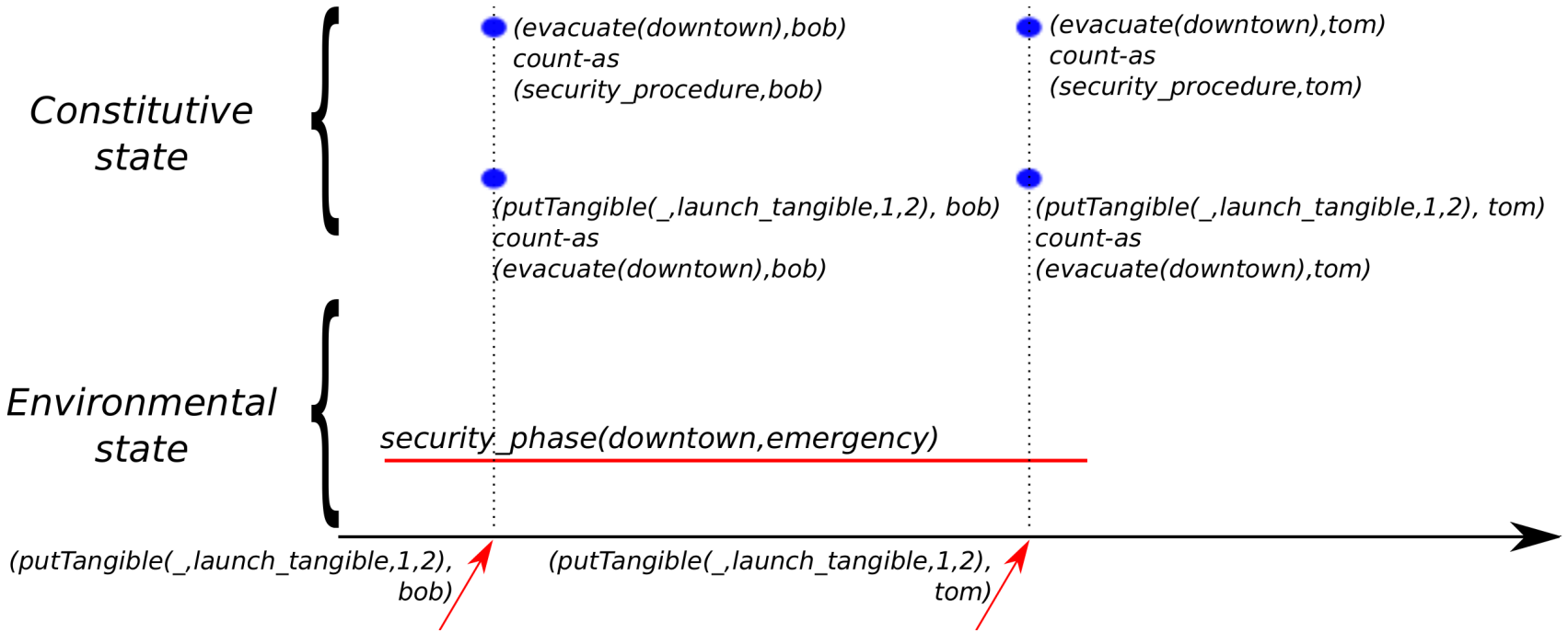
1st order

```
putTangible(_,launch_tangible,1,2)
                    [sai__agent(_)]
  count-as evacuate(downtown)
  while security_phase(downtown,emergency).
```

2nd order

```
evacuate(downtown)
  count-as security_procedure.
```

Constitutive dynamics - example



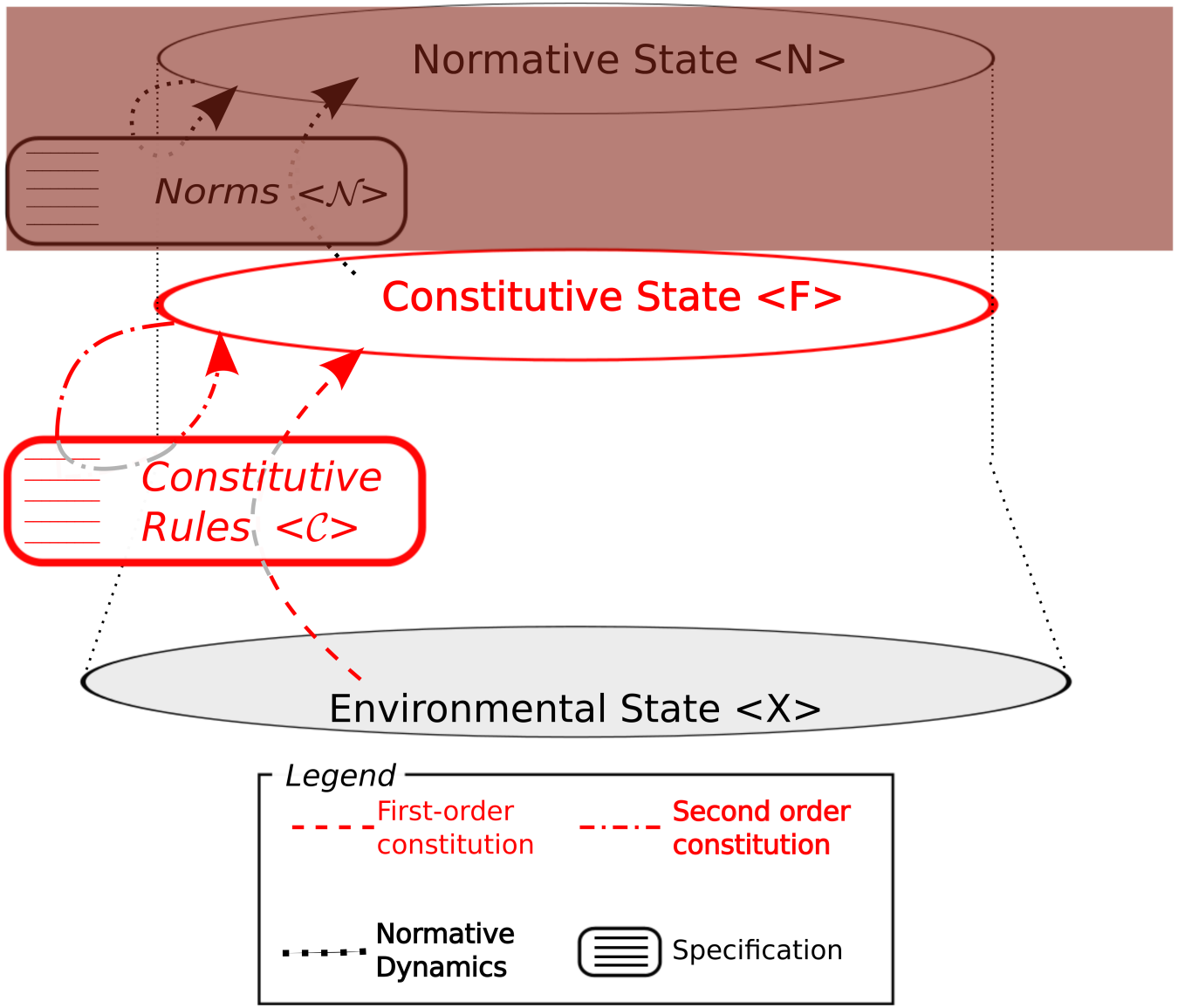
1st order

```
putTangible(_,launch_tangible,1,2)
                    [sai__agent(_)]
  count-as evacuate(downtown)
  while security_phase(downtown,emergency).
```

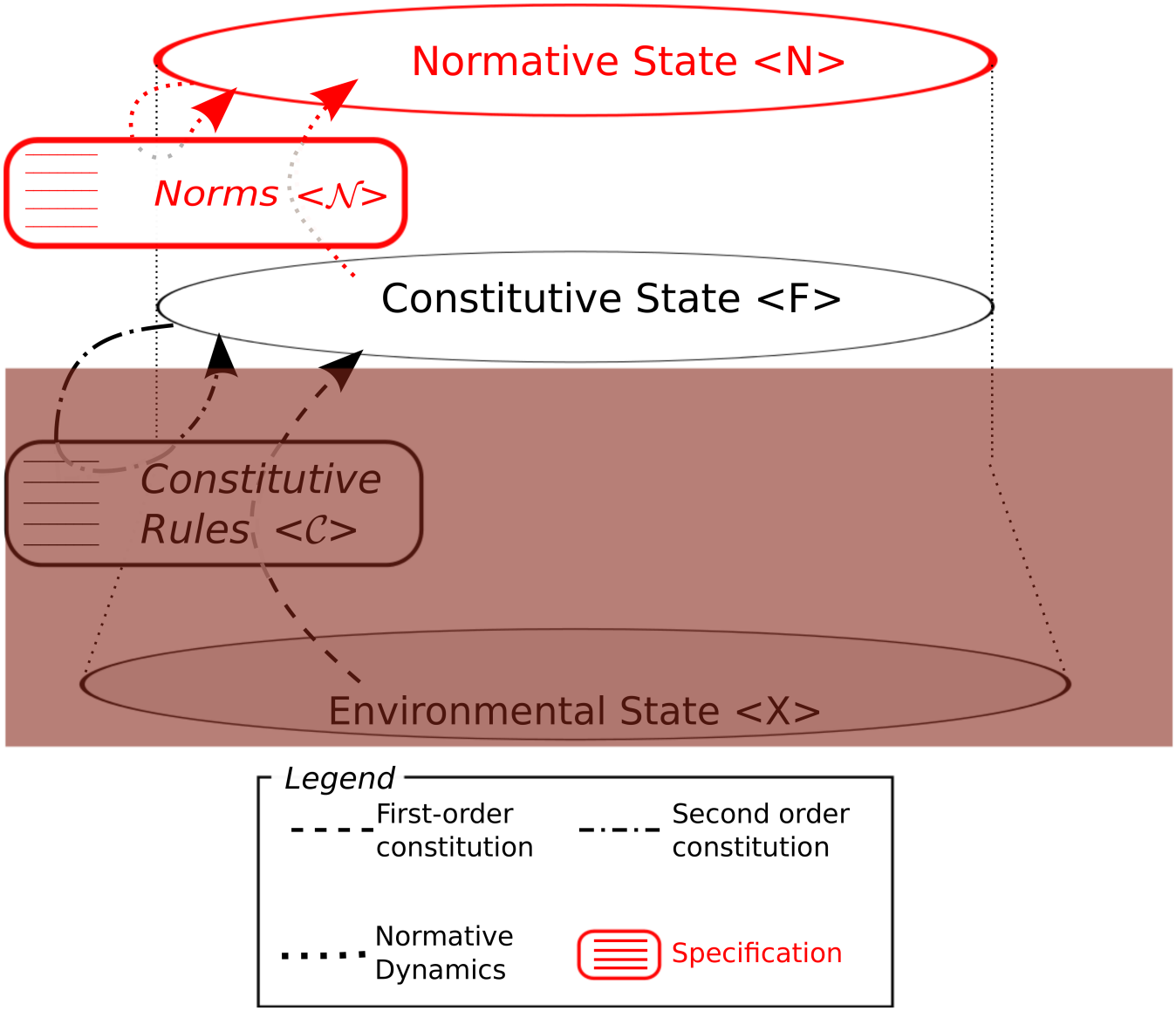
2nd order

```
evacuate(downtown)
  count-as security_procedure.
```

Regulation in SAI



Regulation in SAI



Regulation in SAI

Coupling norms with the constitutive state

(considering their specific representations and dynamics)

Requirements:

1. Normative representation:

   to introduce the status functions in the norm representation

2. Normative dynamics

   to define how the different components of the norms are interpreted

Regulation in SAI

Coupling norms with the constitutive state

(considering their specific representations and dynamics)

Requirements:

1. **Normative representation**:

   to introduce the status functions in the norm representation

2. Normative dynamics

   to define how the different components of the norms are interpreted

Regulation in SAI

Coupling norms with the constitutive state

(considering their specific representations and dynamics)

Requirements:

1. Normative representation:

   to introduce the status functions in the norm representation

2. **Normative dynamics**

   to define how the different components of the norms are interpreted

Norm representation in SAI

Introducing status functions in the norms

A generic approach:

based on the ADICO model [Crawford e Ostrom 1995]

- $\mathbf{A} \in \mathcal{A}_{\mathcal{F}}$
- $\mathbf{I} \in \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$
- $\mathbf{C} \in \mathcal{W}_{\mathcal{F}}$

where

- **A** is the agent targetted by the norm
- **I** is the prescribed outcome
- **C** are the conditions under which the norm must be followed

Normative dynamics with SAI

Different normative models have particular dynamics

Difficult to generalise but...
... all the norms can be at least activated and fulfilled

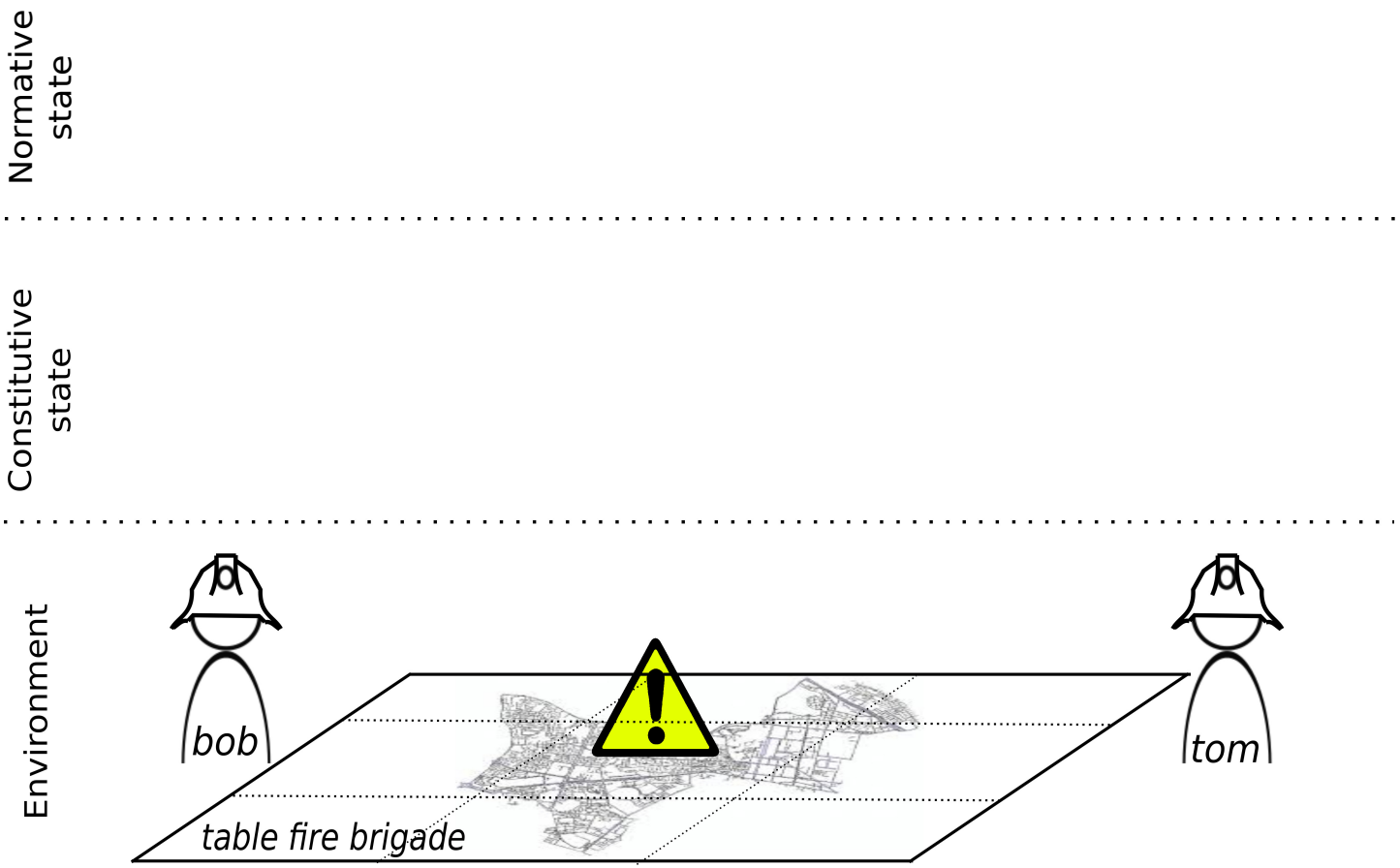
In the sequence: some details on norm activation and fulfilment

considering norms by [Panagiotidi et al., 2013]

Activation

– Norm instances are created when activation condition is satisfied. An instance for every agent carrying α

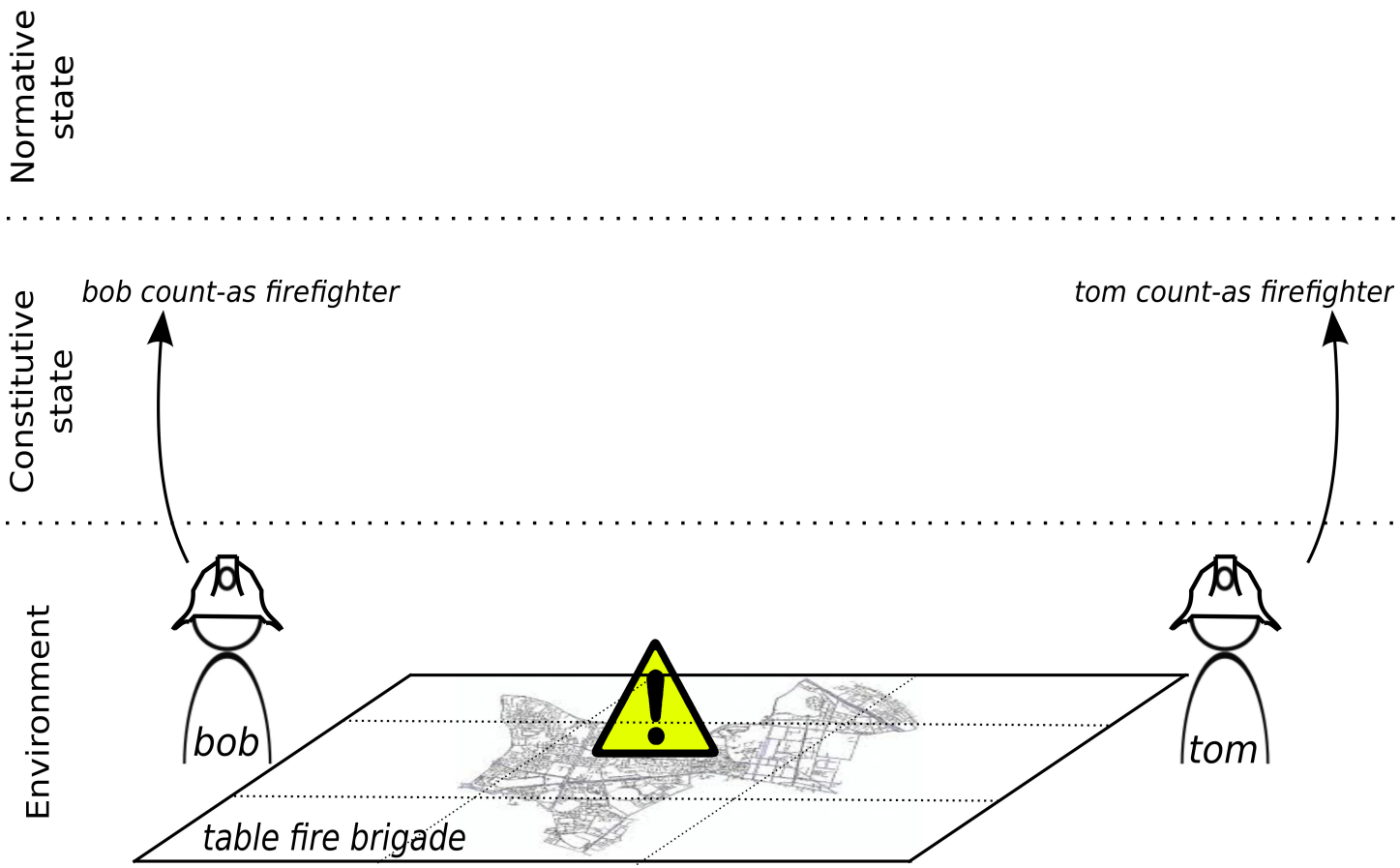
e.g. $\langle firefighter, insecure(Zone), \cdots, evacuate(Zone), \cdots, \cdots \rangle$



Activation

– Norm instances are created when activation condition is satisfied. An instance for every agent carrying α

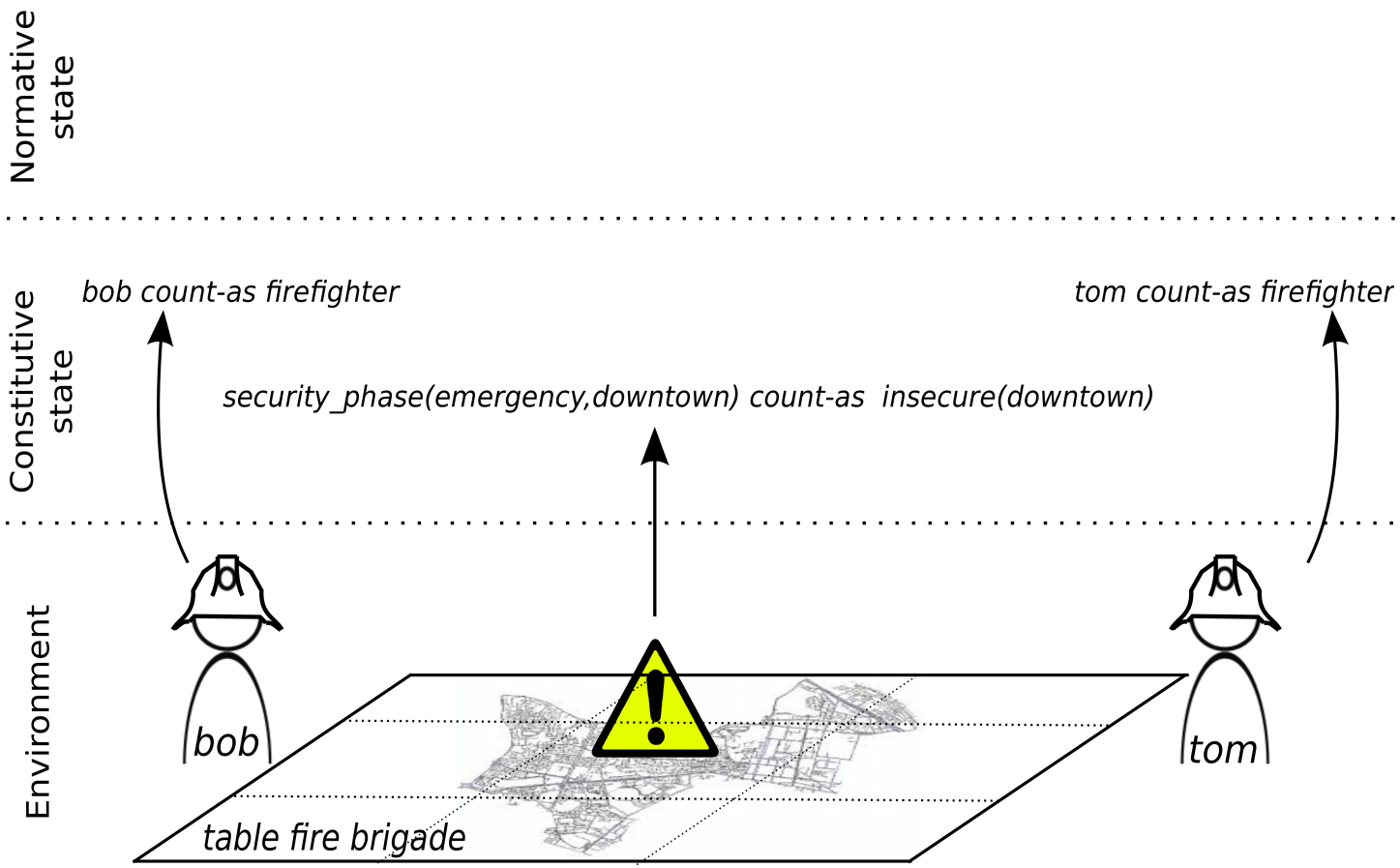
e.g. $\langle \mathit{firefighter}, \mathit{insecure}(\mathit{Zone}), \cdots, \mathit{evacuate}(\mathit{Zone}), \cdots, \cdots \rangle$



Activation

– Norm instances are created when activation condition is satisfied. An instance for every agent carrying α

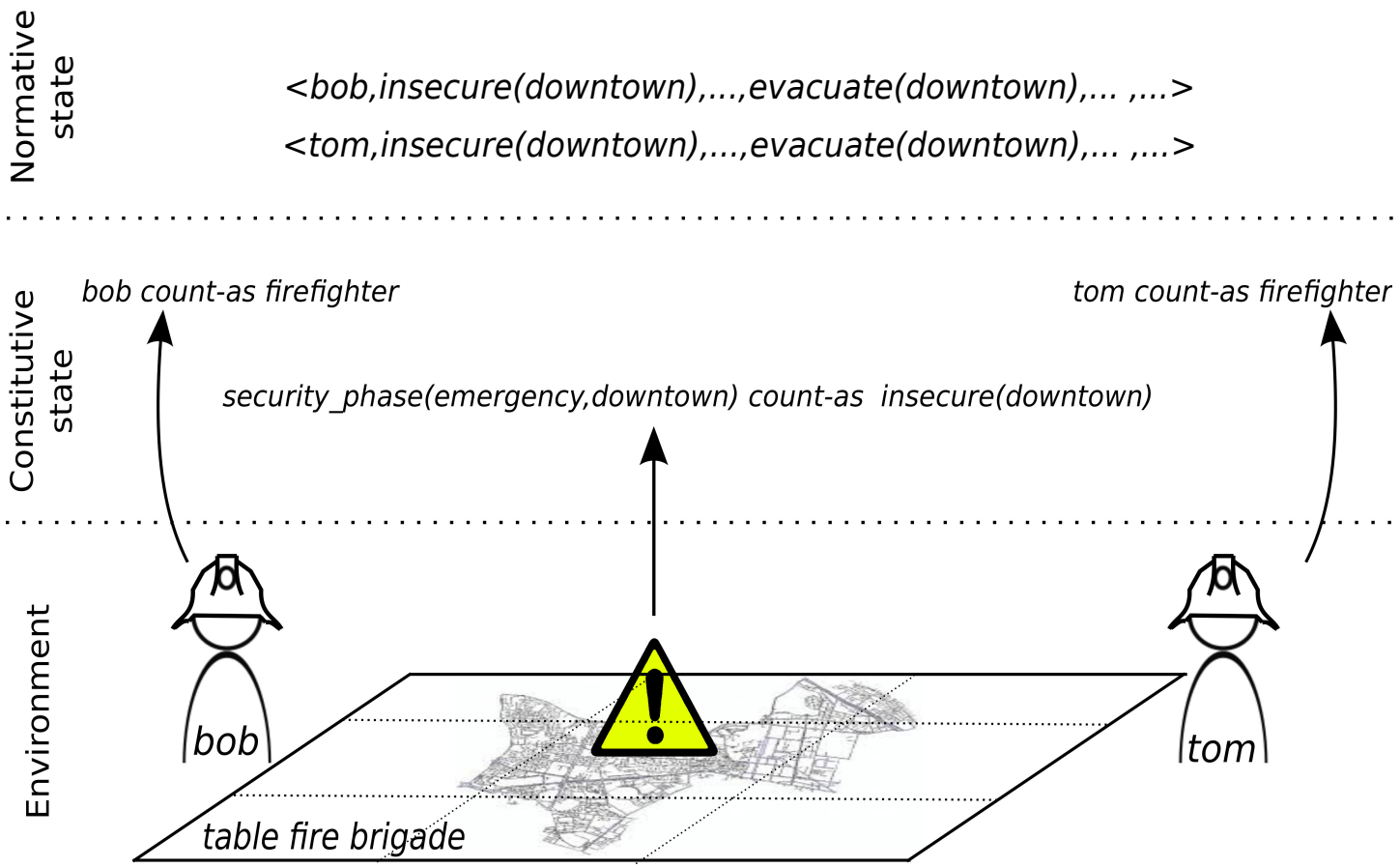
e.g. $\langle firefighter, insecure(Zone), \cdots, evacuate(Zone), \cdots, \cdots \rangle$



Activation

– Norm instances are created when activation condition is satisfied. An instance for every agent carrying α

e.g. $\langle firefighter, insecure(Zone), \cdots, evacuate(Zone), \cdots, \cdots \rangle$



Activation

Formally:

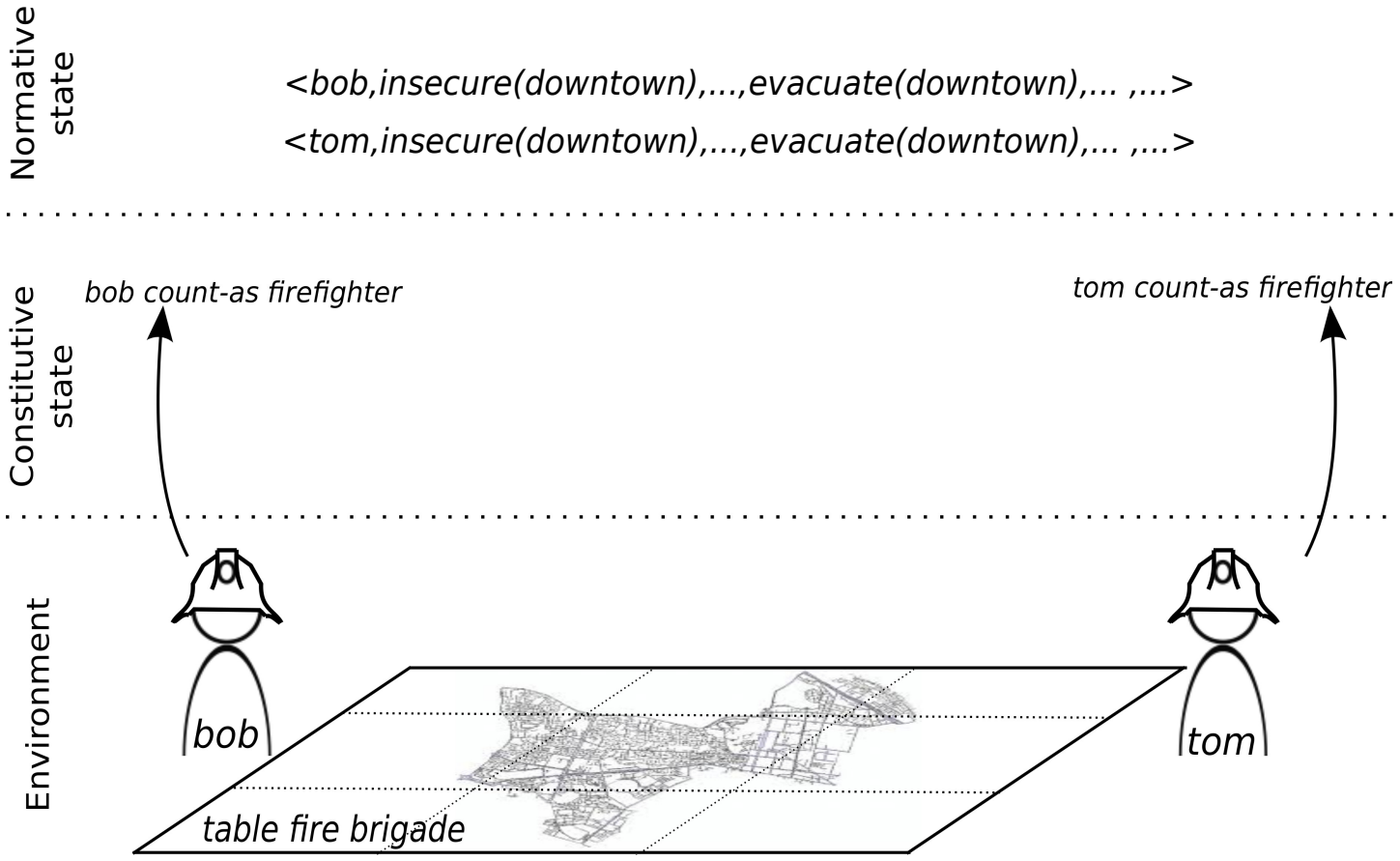
$$activated(\mathcal{N}, F, N) = \{n' | \exists\theta\exists\langle\alpha, c_a, c_m, c_d, c_r, c_t\rangle \in \mathcal{N} : F \cup N \models c_a\theta \wedge (a_X \textbf{ is } \alpha\theta) \wedge n' \notin AS\}$$

$$\text{s.t.} \quad n' = \langle(a_X, \alpha\theta), c_a\theta, c_m\theta, c_d\theta, c_r\theta, c_t\theta\rangle$$

Fulfilments

Individual agent level approach for events

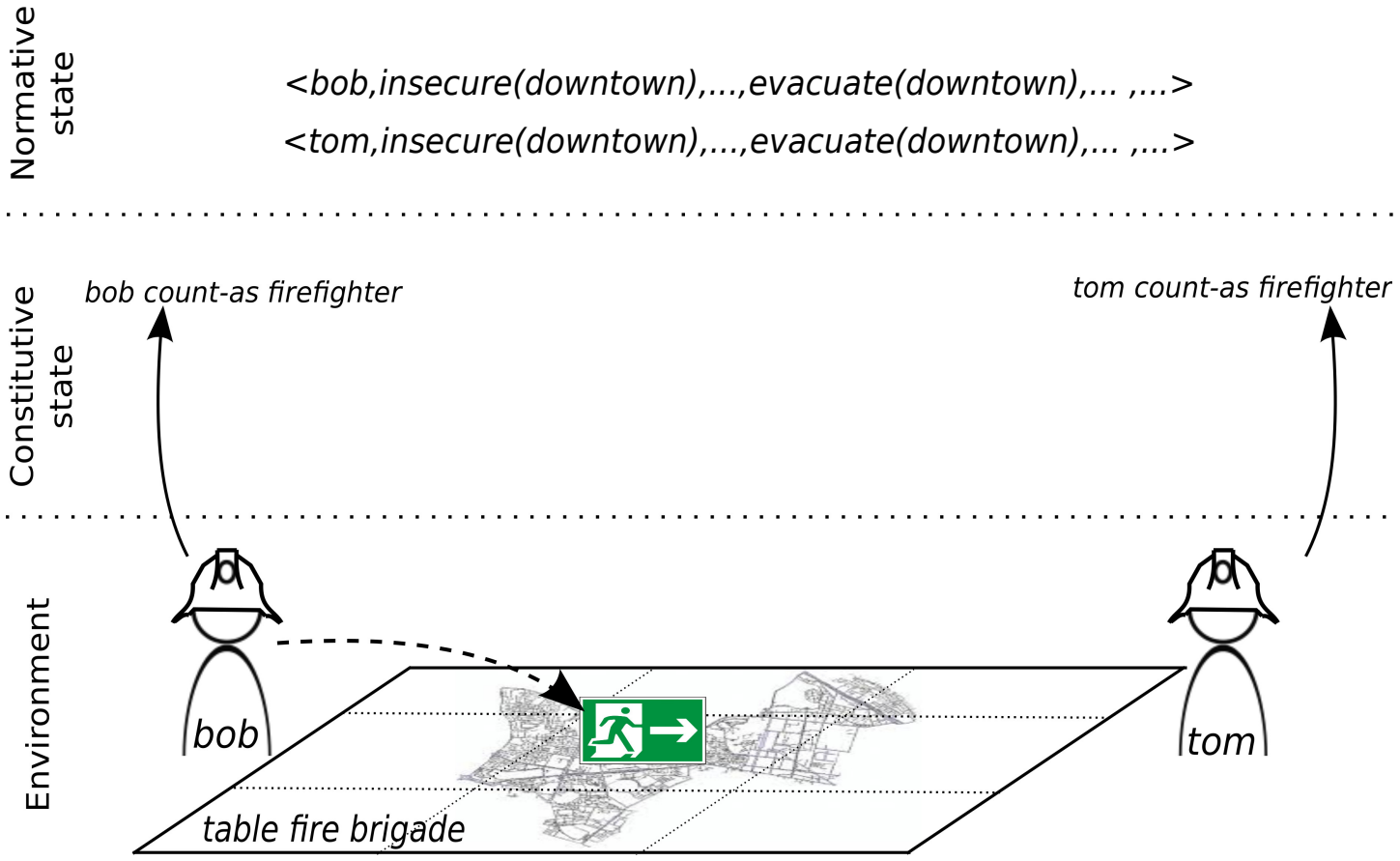
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

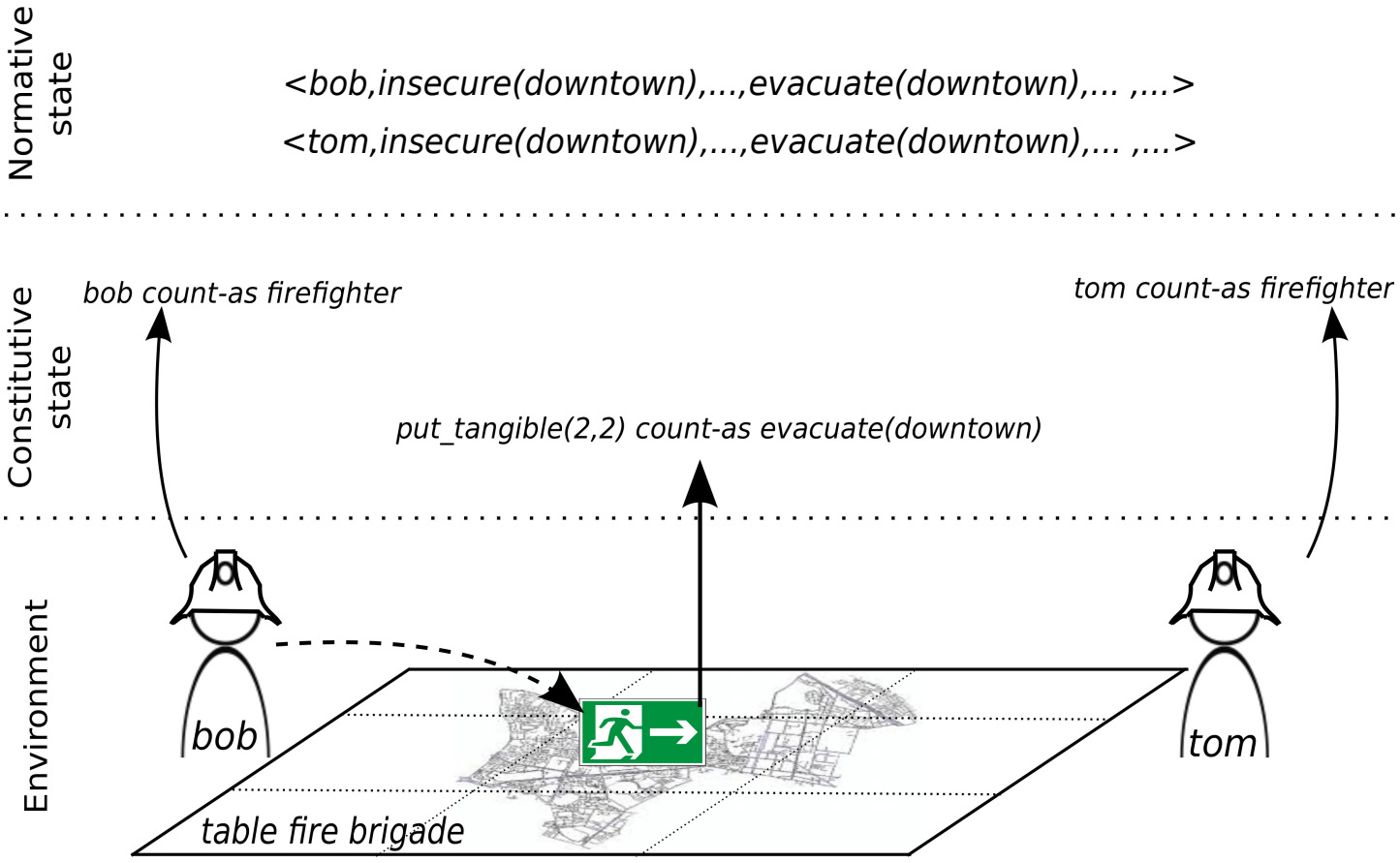
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

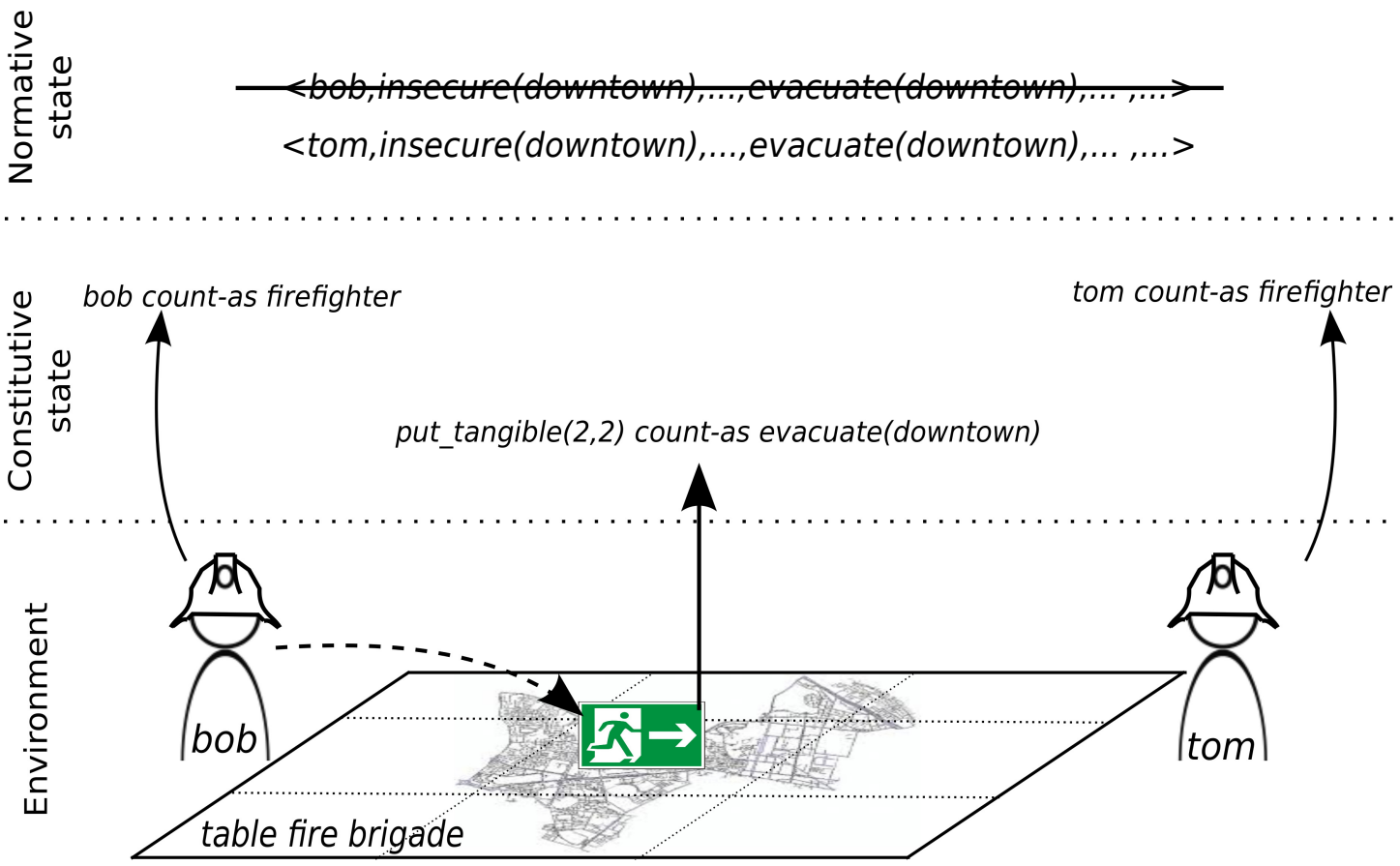
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

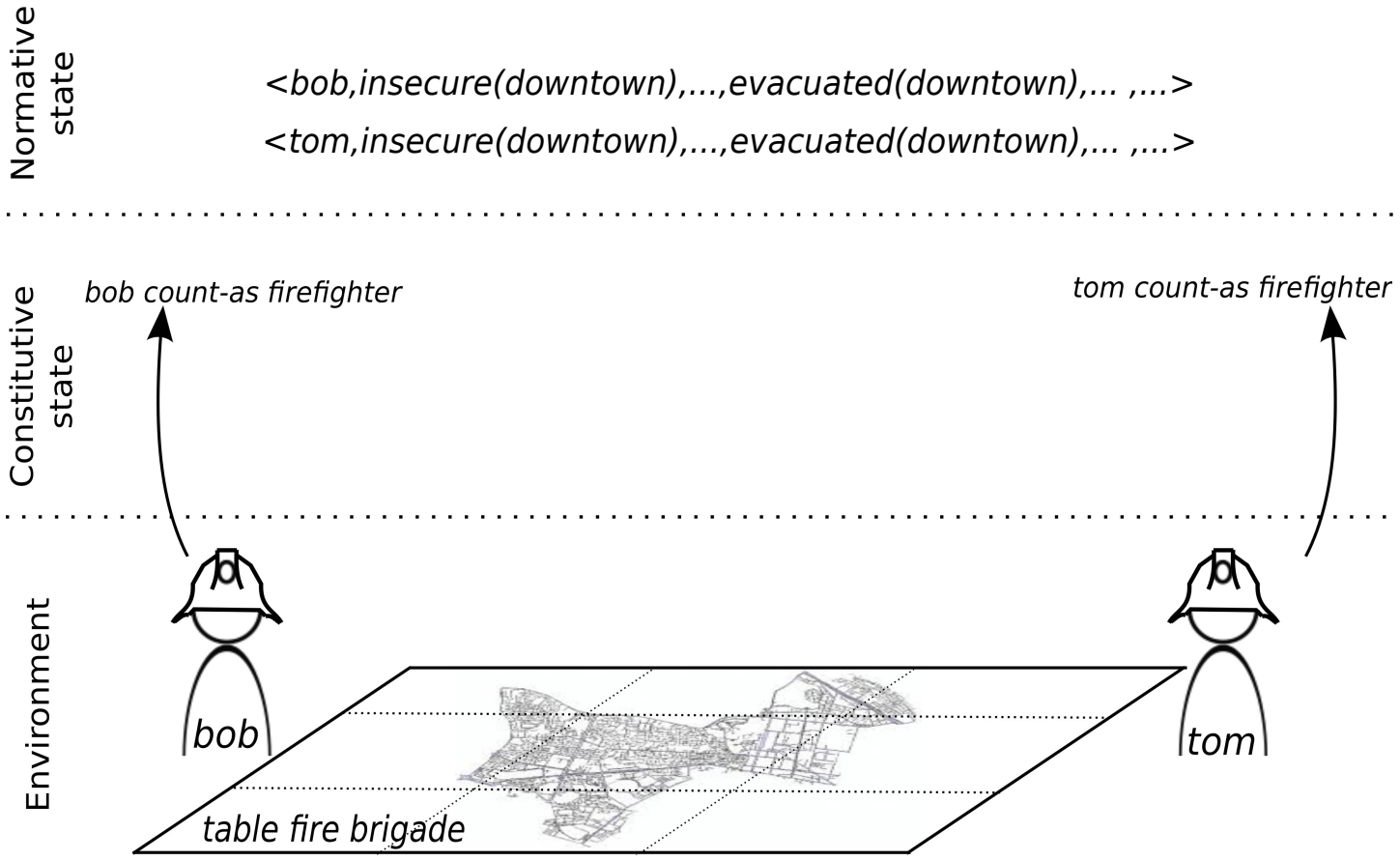
(`evacuate(.)` is an event status function)



Fulfilments

See to it approach for states

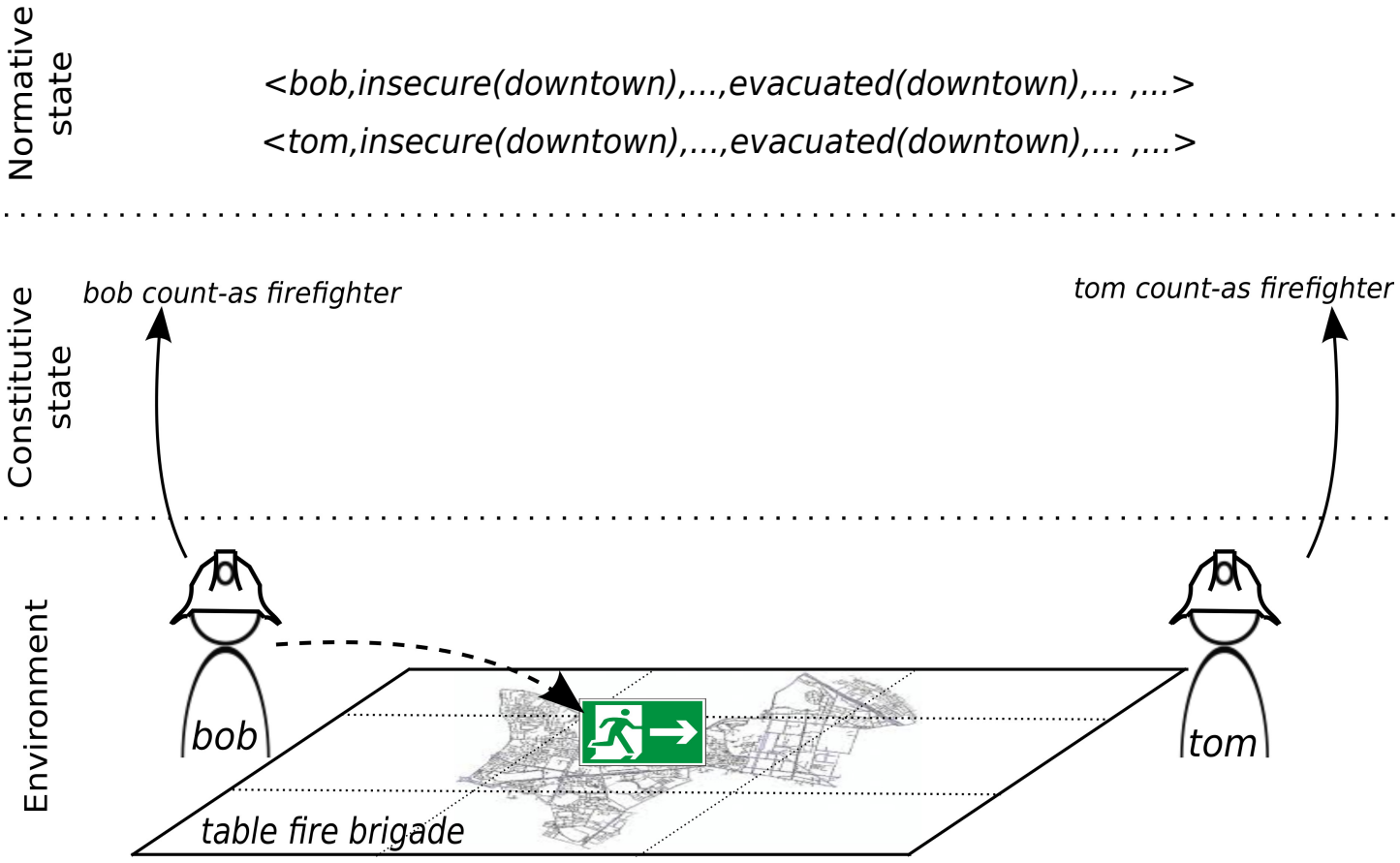
(`evacuated(.)` is a state status function)



Fulfilments

See to it approach for states

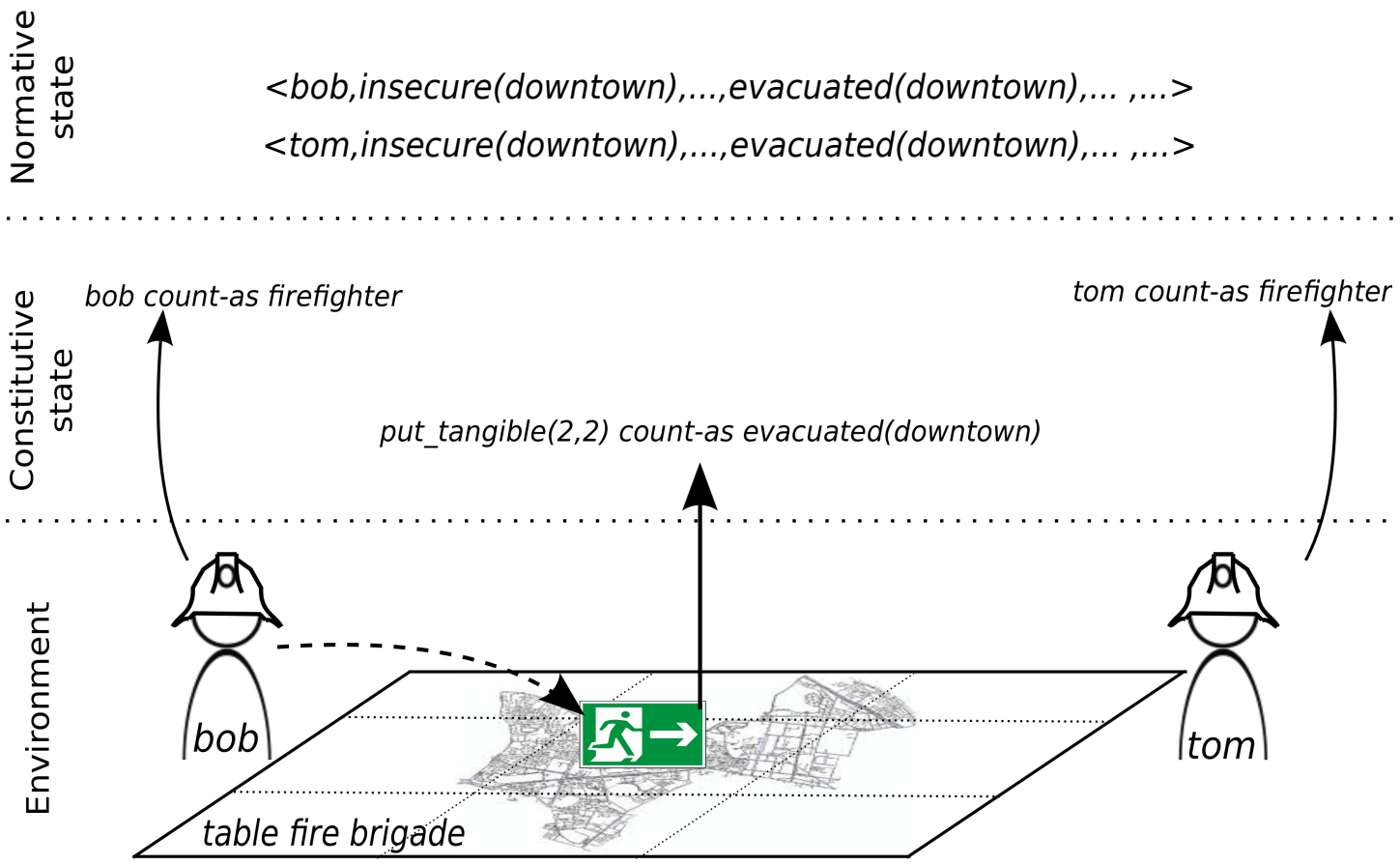
(`evacuated(.)` is a state status function)



Fulfilments

See to it approach for states

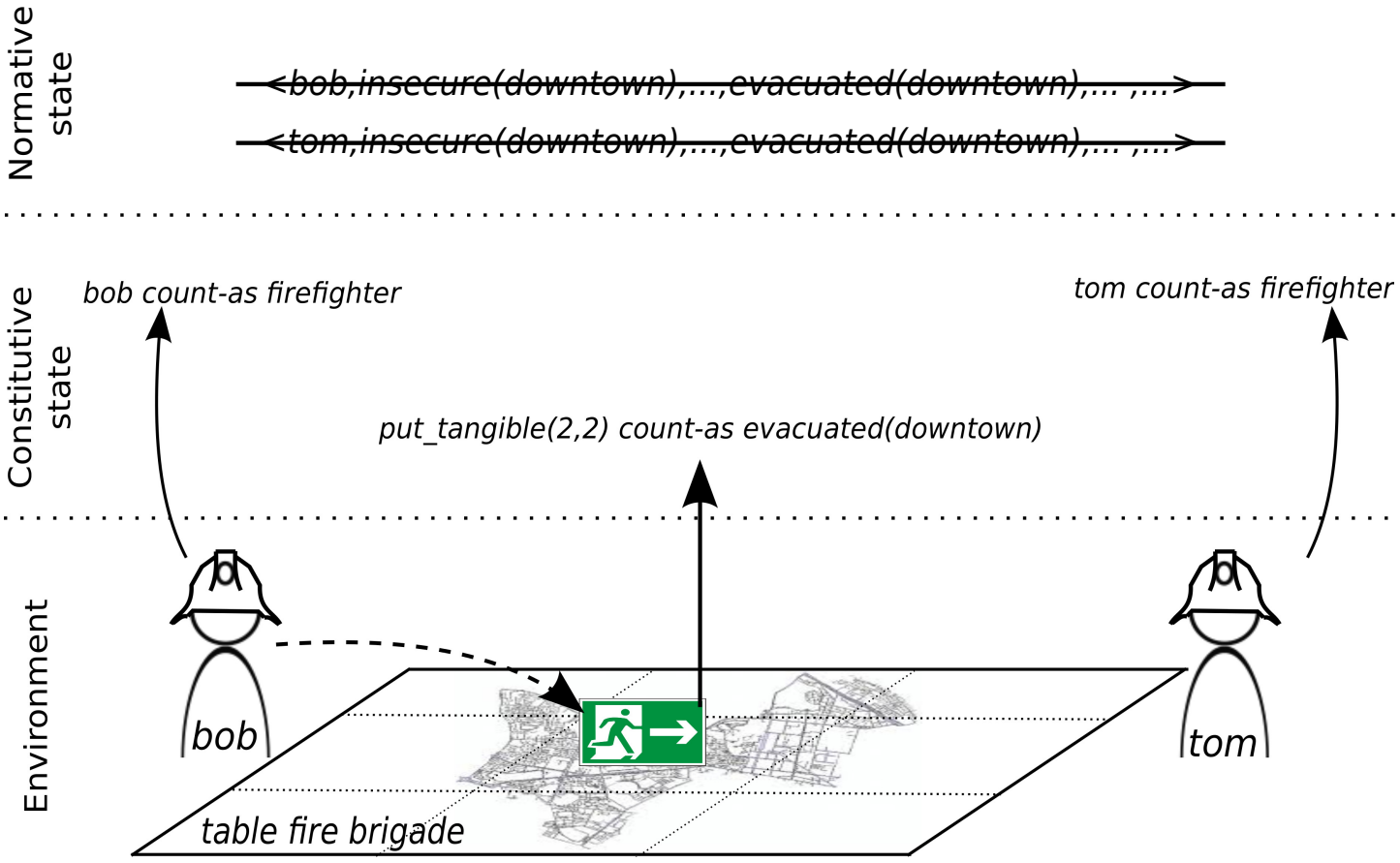
(`evacuated(.)` is a state status function)



Fulfilments

See to it approach for states

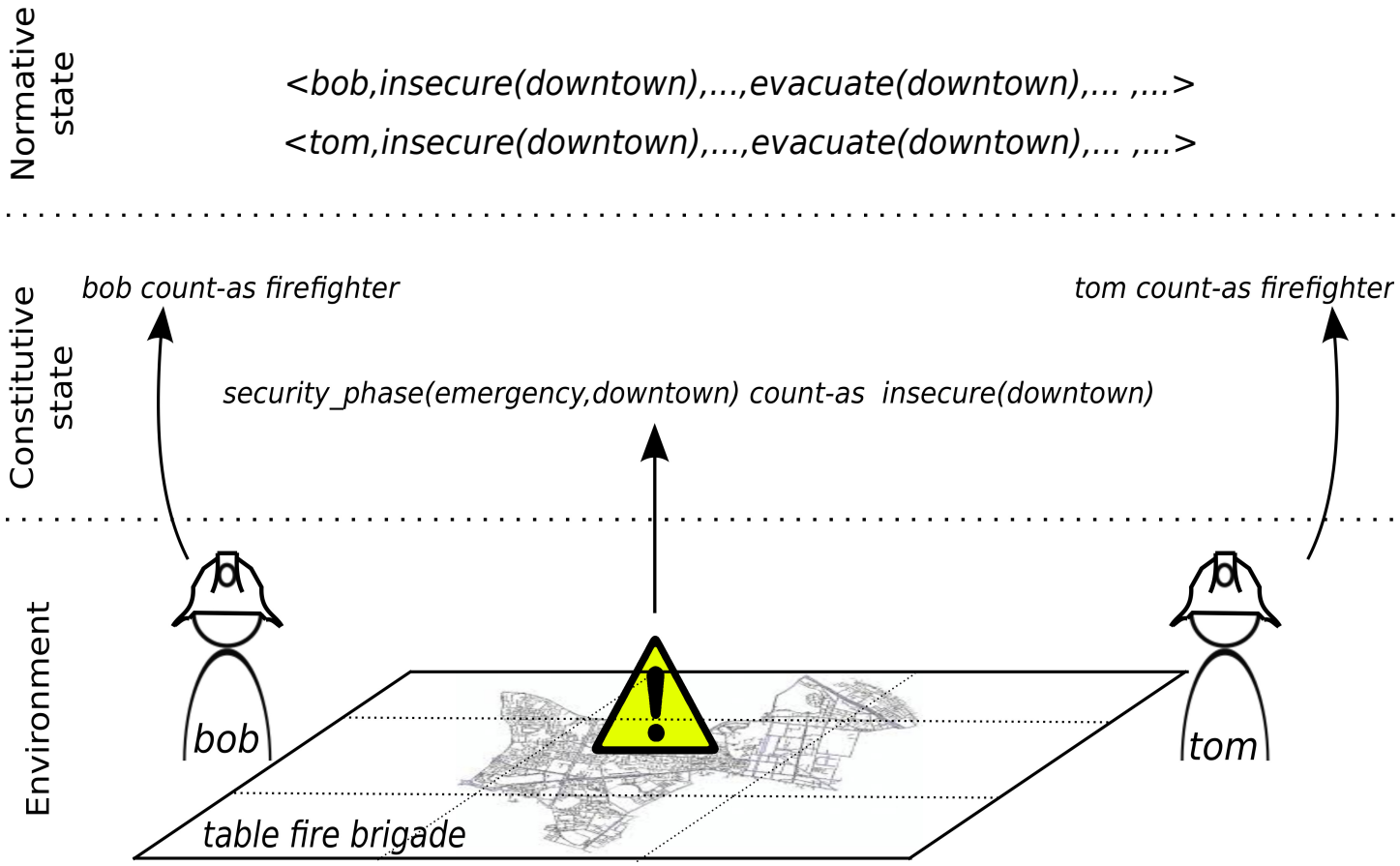
(`evacuated(.)` is a state status function)



Fulfilments

Revocation of agent-status function assignment

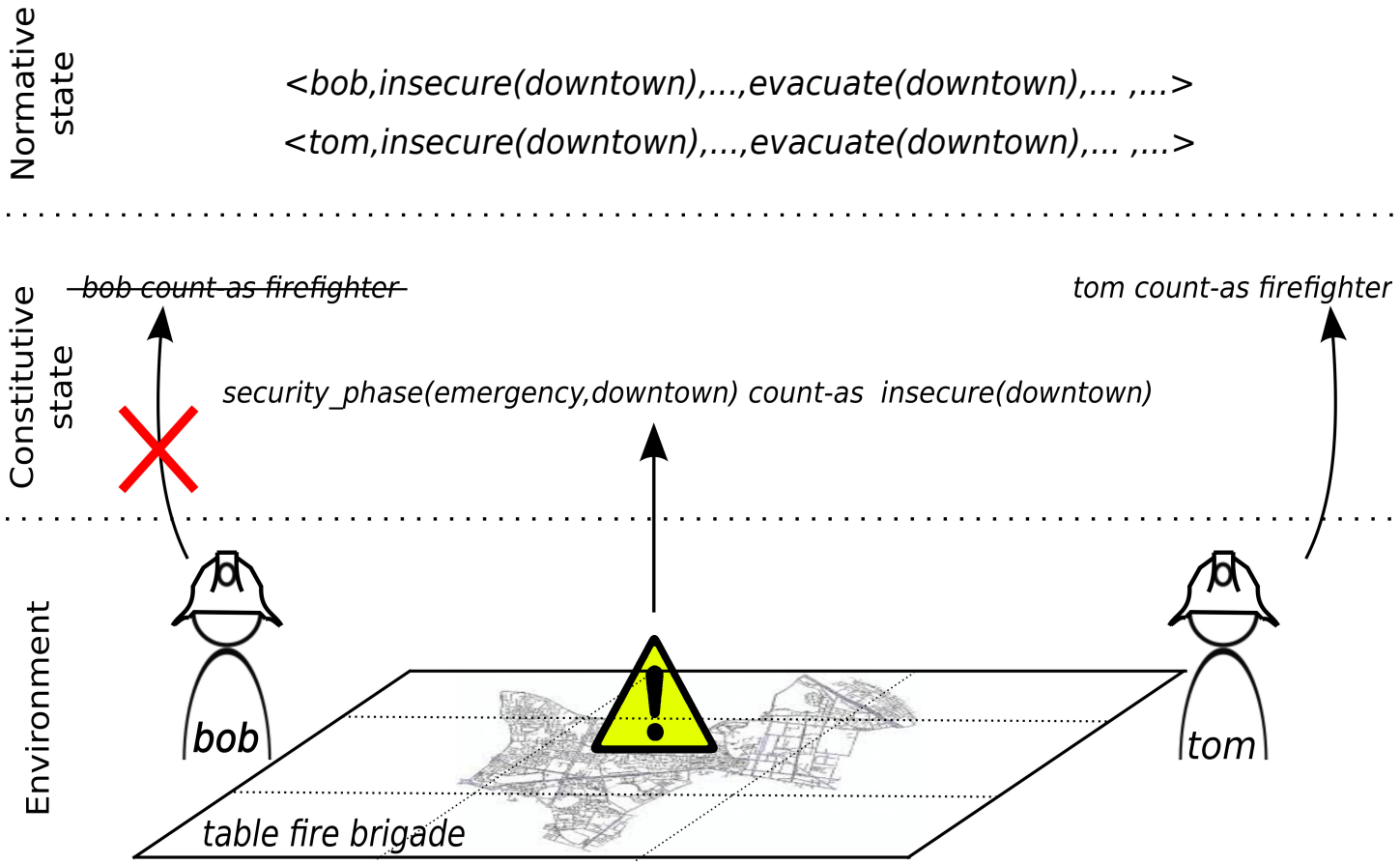
e.g. $\langle firefighter,\ insecure(Zone),\ \cdots,\ evacuate(Zone),\ \cdots,\ \cdots \rangle$



Fulfilments

Revocation of agent-status function assignment

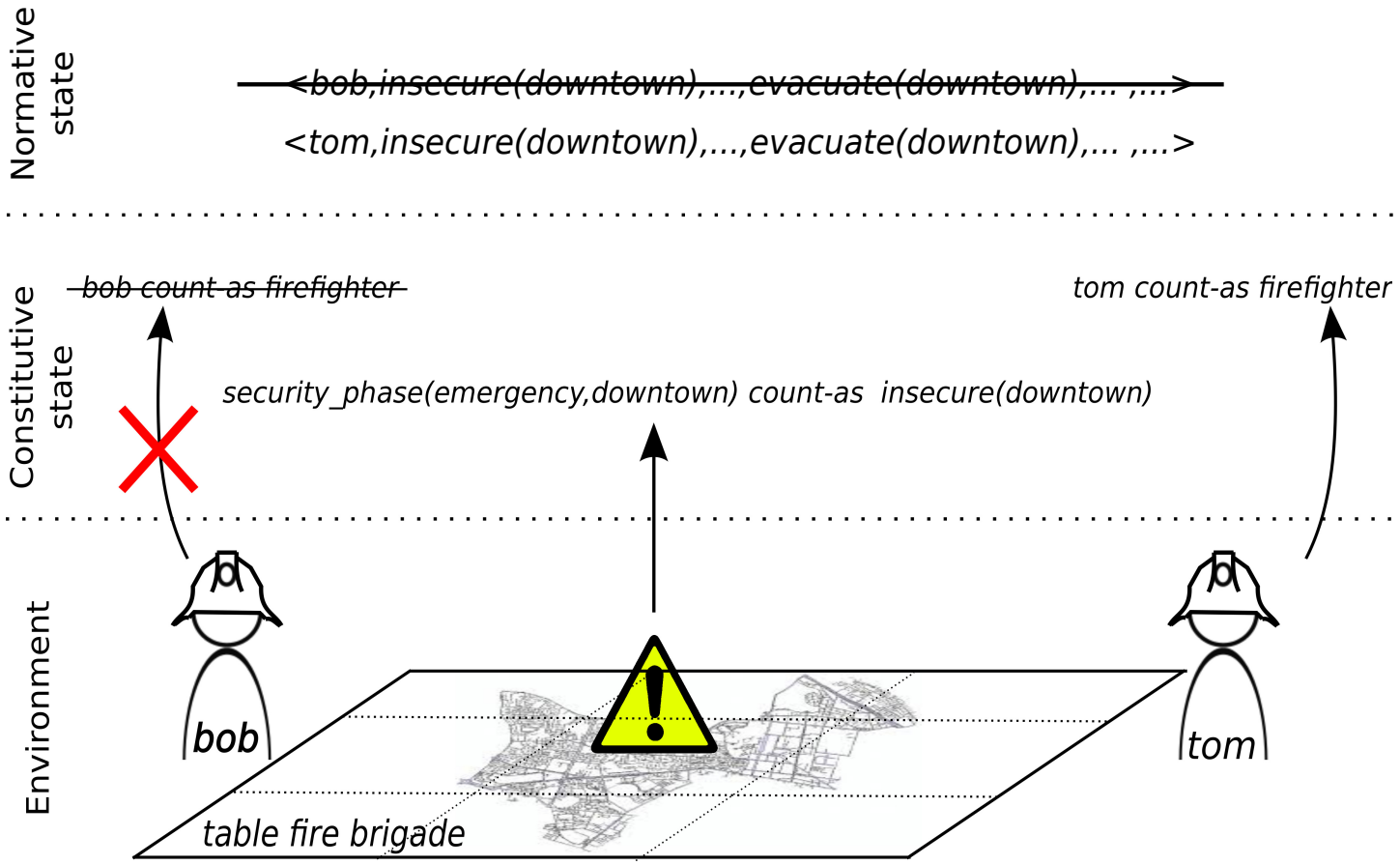
e.g. $\langle firefighter,\ insecure(Zone),\ \cdots,\ evacuate(Zone),\ \cdots,\ \cdots \rangle$



Fulfilments

Revocation of agent-status function assignment

e.g. $\langle firefighter,\ insecure(Zone),\ \cdots,\ evacuate(Zone),\ \cdots,\ \cdots \rangle$



Fulfilments

Formally:

$$f\text{-}deactivated^e(F,N) = \{\langle n' | \exists (e_X, a_X) \in E_X : n' \in AS \wedge c'_d \in \mathcal{E}_{\mathcal{F}} \wedge F \models ((e_X, a_X) \text{ is } c'_d \vee \neg(a_X \text{is } \alpha)) \wedge F \cup N \models c'_m\}$$

$$f\text{-}deactivated^s(F,N) = \{\langle n' | n' \in AS \wedge c'_d \in \mathcal{S}_{\mathcal{F}} \wedge F \models (c'_d \vee \neg(a_X \text{is } \alpha)) \wedge F \cup N \models c'_m\}$$

$$r\text{-}deactivated^e(F,N) = \{n' | \exists (e_X, a_X) \in E_X : n' \in VS \wedge c'_r \in \mathcal{E}_{\mathcal{F}} \wedge F \models ((e_X, a_X) \text{ is } c'_r \vee \neg(a_X \text{is } \alpha))\}$$

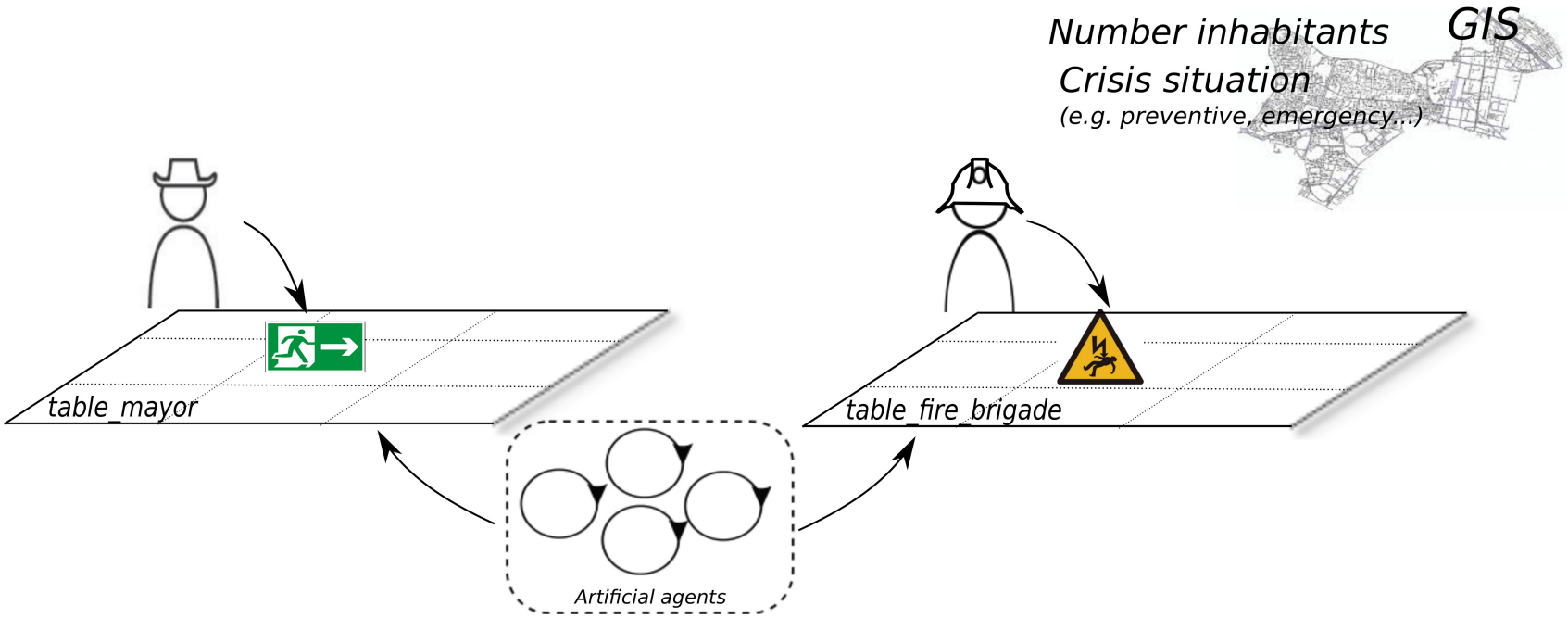
$$r\text{-}deactivated^s(F,N) = \{n' | n' \in VS \wedge c'_r \in \mathcal{S}_{\mathcal{F}} \wedge F \models (c'_r \vee \neg(a_X \text{is } \alpha))\}$$

$$\text{s.t.} \quad n' = \langle (a_X, \alpha), c'_a, c'_m, c'_d, c'_r, c'_t \rangle$$

Application example: crisis management

Collaborative System [Thévin et al. 2014]

- Tangible: TangiSense tables [Kubicki et al. 2012]
- Normative: regulated by norms
- Requirement: to institutionalise facts occurring in the environment



Application example: crisis management

```
/* The firefighter is prohibited to evacuate secure zones */
n1: secure(Zone)
   -> prohibition(firefighter,n1,evacuate(Zone),‘now’+‘1 day’).

/*  The firefighter is obliged to evacuate insecure zones */
n2: insecure(Zone)
   -> obligation(firefighter,n2,evacuate(Zone),‘now’+‘1 day’).
```

Agent Status functions: **firefighter**

Event Status functions: **evacuate(Zone)**

State Status functions: **secure(Zone)**, **insecure(Zone)**

Application example: crisis management

```
/* The firefighter is prohibited to evacuate secure zones */
n1: secure(Zone)
   -> prohibition(firefighter,n1,evacuate(Zone),‘now’+‘1 day’).

/*  The firefighter is obliged to evacuate insecure zones */
n2: insecure(Zone)
   -> obligation(firefighter,n2,evacuate(Zone),‘now’+‘1 day’).
```

```
/* Actors carry the status functions
   according to their check in the tables */
 Agent count-as firefighter
     when checkin(table_fire_brigade,Agent).
```

Application example: crisis management

```
/* The firefighter is prohibited to evacuate secure zones */
n1: secure(Zone)
   -> prohibition(firefighter,n1,evacuate(Zone),'now'+'1 day').

/*  The firefighter is obliged to evacuate insecure zones */
n2: insecure(Zone)
   -> obligation(firefighter,n2,evacuate(Zone),'now'+'1 day').
```

```
/* A zone preventive phase of crisis managament counts as
   that zone being secure if (i) it does not pose
   electrical risks and (ii) it has at most 500 inhabitants */
security_phase(_,Zone,preventive) count-as secure(Zone)
   while not( AnyState is electric_risky(Zone)) &
         ((nbInhabit(_,Zone,X)& X<=500) |
         security_phase(_,Zone,preventive) is secure(Zone)).
```

Application example: crisis management

```
/* The firefighter is prohibited to evacuate secure zones */
n1: secure(Zone)
   -> prohibition(firefighter,n1,evacuate(Zone),‘now’+‘1 day’).

/*  The firefighter is obliged to evacuate insecure zones */
n2: insecure(Zone)
   -> obligation(firefighter,n2,evacuate(Zone),‘now’+‘1 day’).
```

```
/* Firefighter putting the object launch_tangible in the
   coordinates 1,2 of a table counts as the evacuation of the downtown */
putTangible(_,launch_tangible,1,2)[sai__agent(Actor)]
   count-as evacuate(downtown)
   while Actor is firefighter.

putTangible(_,launch_tangible,3,3)[sai__agent(Actor)]
   count-as evacuate(industrial_zone)
   while Actor is firefighter.
```

Conclusions

Contributions of SAI

1. Institutional reality becomes more explicit and external

2. Unified representation of institutional reality
   shared among different normative models
   wide normative coverage

3. Independent normative and constitutive layers
   particular but connected normative and constitutive models

Conclusions

Contributions of SAI

4. Agents can reason about the normative consequences of their acts
   Norms explicitly refer to status functions whose constitution is also explicit

5. Consistent institutional programming
   Norms must refer to status functions and the constitution must produce status function assignments

6. Regulation consistent with environment
   Norms take status functions as having a specific nature that is taken into account in the constitutive management

7. Consistent regulative outcomes
   Inconsistencies come from problems in the normative specification instead of from the link with the environment

What is missing - Future work

- Designing object-status functions
- Institutionalising other social abstractions other than norms
- Methodology to specify institutions

Final remarks

SAI available at

`github.com/artificial-institutions/sai`

constitutive language interpreter, interfaces for normative platforms, examples, etc.

SAI vs. JaCaMo - initial integration

```
mas house_building {
   agent giacomo
   agent companyA
   agent companyB

   workspace auction{
      artifact auction_for_SitePreparation: AuctionArt("SitePreparation", 2000)
      artifact auction_for_Floors: AuctionArt("Floors",1000)
      artifact auction_for_Walls: AuctionArt("Walls", 1000)
                              ⋮
   }
   institution house_institution: constitutive.sai{
       workspaces: auction
   }
   organisation house_organisation : house-os.xml {
      group house_group: house_group
      scheme house_scheme: build_house_sch
      situated: house_institution
   }

}
```

SAI vs. JaCaMo - initial integration

```
// obligation to achieve a goal
// ============================
// the agent perceives the obligation following the NPL notation
+obligation(Ag,R,done(Scheme,Goal,Ag),Deadline):
   .my_name(Ag) &
   //The agent looks for a constitutive rule defining how the goal is achieved
   constitutive_rule(X,done(Scheme,Goal,Ag),ToDo[sai__agent(Ag)],M)
   <- println(``I am obliged to'' ,Goal,``.'',
              ``I found a constitutive rule that shows me''
               I have to produce the event '', ToDo);
      focus(House);
      ToDo[artifact_id(House)].
```

Publications I

1. BRITO, M. de; HÜBNER, J. F. **Institutional Situatedness in Multi-Agent Systems**. WESAAC 2014. Porto Alegre - Brazil.

2. BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. **A conceptual model for situated artificial institutions**. CLIMA XV. 2014. Prague - Czech Republic.

3. BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. **Bringing constitutive dynamics to situated artificial institutions**. EPIA 2015. Coimbra - Portugal.

4. BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. **Coupling regulative and constitutive dimensions in situated artificial institutions**. EUMAS 2015. Athens - Greece.

Publications II

5. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Situated artificial institution to support advanced regulation in the field of crisis management**. PAAMS 2015. Salamanca - Spain.

6. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Situated regulation on a crisis management collaboration platform**. PAAMS 2015. Salamanca - Spain.

7. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Institution artificielle située pour une aide à la régulation dans le cadre de la gestion de crises**. JFMSMA 2015. Rennes - France.

8. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Supporting flexible regulation of crisis management by means of situated artificial institutions**. Frontiers of Information Technology & Electronic Engineering (FITEE), v.17, n.4.

Publications III

9. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Institution artificielle située pour une aide à la régulation de la gestion de crises**. Revue d'Intelligence Artificielle (RIA), v.30, n.1-2.

Submission

1. BRITO, M. de; HÜBNER, J. F; BOISSIER, O. **Situated Artificial Institutions: Stability, consistency, and flexibility in the regulation of agent societies**. JAAMAS.

References I

Huib Aldewereld, Sergio Álvarez Napagao, Frank Dignum, and Javier Vázquez-Salceda.
Making norms concrete.
In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen, editors, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010): Volume 1-3*, pages 807–814. IFAAMAS, 2010.

Maiquel de Brito, JomiFred Hübner, and Olivier Boissier. A conceptual model for situated artificial institutions. *Computational Logic in Multi-Agent Systems*, volume 8624 of *LNCS*. 2014.

Guido Boella and Leendert van der Torre.
A logical architecture of a normative system.
In Lou Goble and John-Jules Ch. Meyer, editors, *Deontic Logic and Artificial Normative Systems (DEON 2006)*, volume 4048 of *LNCS*, pages 24–35. Springer, 2006.

Guido Boella and Leendert W. N. van der Torre.
Regulative and constitutive norms in normative multiagent systems.
In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 255–266. AAAI Press, 2004.

References II

Cristiano Castelfranchi
Engineering Social Order
In: Omicini A., Tolksdorf R., *Zambonelli F. (eds) Engineering Societies in the Agents World. ESAW 2000*. Lecture Notes in Computer Science, vol 1972. Springer, Berlin, Heidelberg

COMMONS, J. R.
Institutional Economics – Its Place in Political Economy. [S.l.]: The Macmillan Company, 1934. (Institutional Economics Series, v. 1).

CRAWFORD, S. E. S.; OSTROM, E. A Grammar of Institutions.
The American Political Science Review, v. 89, n. 3, p. 582–600, 1995.

Mehdi Dastani, Davide Grossi, John-Jules Ch. Meyer, and Nick A. M. Tinnemeier.
Normative multi-agent programs and their logics.
In Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, editors, *Normative Multi-Agent Systems*, volume 09121 of *Dagstuhl Seminar Proceedings*, 2009.

References III

Dastani, M., van der Torre, L., Yorke-Smith, N.
Commitments and interaction norms in organisations
Autonomous Agents and Multi-Agent Systems (2017) 31:2017.

Dignum, Virginia
A model for organizational interaction: based on agents, founded in logic
PhD Thesis, 2004

Dignum, Virginia and Aldewereld, Huib and Dignum, Frank
On the Engineering of Multi Agent Organizations
In Weyns, Danny and Müller, Jörg P, editors, *12th International Workshop on Agent-Oriented Software Engineering (AOSE@AAMAS2011)*

Geiger, R. Stuart
The Lives of Bots
Lovink, Geert and Tkacz, Nathaniel, editors, *Critical Point of View: A Wikipedia Reader*,Institute of Network Cultures,2011.

HODGSON, G. M. What are institutions.
Journal of Economic Issues, n. 1, p. 1–25, 2006.

References IV

Jomi F. Hubner, Jaime S. Sichman, and Olivier Boissier
Developing organised multiagent systems using the MOISE$^+$ model: programming issues at the system and agent levels
Int. J. Agent-Oriented Softw. Eng. 1, 3/4 (December 2007), 370-395.

MILLER, S.
Social institutions.
In: Zalta, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Fall 2012. [S.l.: s.n.], 2012.

NORTH, D. C. Economic performance through time.
American Economic Review, v. 84, p. 359–368, 1994.

SEARLE, J. The Construction of Social Reality. [S.l.]: Free Press, 1995.

SEARLE, J. Making the Social World:The Structure of Human Civilization. [S.l.]: Oxford University Press, 2009.

References V

Shoham, Y., Tennenholtz, M.
On the synthesis of useful social laws for artificial agent societies(preliminary report).
In Proceedings of the 10th national conference on artificial intelligence. San Jose, CA, July 12–16, 1992 (pp. 276–281).

Munindar P. Singh
An Ontology for Commitments in Multiagent Systems
Artificial Intelligence and Law 7:97.

Sofia Panagiotidi, Sergio Álvarez-Napagao, and Javier Vázquez-Salceda.
Towards the norm-aware agent: Bridging the gap between deontic specifications and practical mechanisms for norm monitoring and norm-aware planning.
In *COIN IX*, pages 346–363, 2013.

LUCK, M. et al. Agent technology: computing as interaction (a roadmap for agent based computing). University of Southampton, 2005.

BRITO, M. de; HÜBNER, J. F.; BORDINI, R. H. Programming institutional facts in multi-agent systems. In: ALDEWERELD, H.; SICHMAN, J. S. a. (Ed.). Coordination, Organizations, Institutions, and Norms in Agent Systems VIII. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 7756). p. 158–173.

References VI

DASTANI, M.; TORRE, L. W. N. van der; YORKE-SMITH, N. Monitoring interaction in organisations. In: Coordination, Organizations, Institutions, and Norms in Agent Systems VIII - 14th International Workshop, COIN 2012, Held Co-located with AAMAS 2012, Valencia, Spain, June 5, 2012, Revised Selected Papers. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 17–34.

BOELLA, G.; TORRE, L. van der. Constitutive norms in the design of normative multiagent systems. In: TONI, F.; TORRONI, P. (Ed.). Computational Logic in Multi-Agent Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 3900). p. 303–319. ISBN 978-3-540-33996-0.

CLIFFE, O.; VOS, M. D.; PADGET, J. Answer set programming for representing and reasoning about virtual institutions. In: INOUE, K.; SATOH, K.; TONI, F. (Ed.). Computational Logic in Multi-Agent Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4371). p. 60–79.

HÜBNER, J. F.; BOISSIER, O.; BORDINI, R. H. A normative programming language for multi-agent organisations. Annals of Mathematics and Artificial Intelligence, v. 62, n. 1-2, 2011.

References VII

Sébastien Kubicki, Sophie Lepreux, and Christophe Kolski. Rfid-driven situation awareness on tangisense, a table interacting with tangible objects. *Personal and Ubiquitous Computing*, 16(8):1079–1094, 2012

Lauren Thévin, Fabien Badeig, Julie Dugdale, Olivier Boissier, and Catherine Garbay. Un système multi-agent normatif pour la collaboration et l'interaction mixte. In *JFSMA 18*, pages 1–10, 2014.

CRAWFORD, S. E. S.; OSTROM, E. A Grammar of Institutions. The American Political Science Review, v. 89, n. 3, p. 582–600, 1995.

LÓPEZ, F. L. y; LUCK, M.; D'INVERNO, M. A normative framework for agent-based systems. In: BOELLA, G.; TORRE, L. W. N. van der; VERHAGEN, H. (Ed.). Normative Multi-agent Systems. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. (Dagstuhl Seminar Proceedings, v. 07122).

PIUNTI, M. et al. Embodied organizations: a unifying perspective in programming agents, organizations and environments. In: Proceedings of The Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW 2010). [S.l.]: CEUR-WS.org, 2010. (CEUR Workshop Proceedings, v. 627).

References VIII

- CAMPOS, J. et al. Formalising Situatedness and Adaptation in Electronic Institutions. In: HÜBNER, J. et al. (Ed.). Coordination, Organizations, Institutions and Norms in Agent Systems IV. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5428). p. 126–139.
- DASTANI, M. et al. Knowledge Representation for Agents and Multi-Agent Systems. In: MEYER, J.-J. C.; BROERSEN, J. (Ed.). Berlin, Heidelberg: Springer-Verlag, 2009. cap. Normative Multi-agent Programs and Their Logics, p. 16–31.
- ALDEWERELD, H. et al. Making norms concrete. In: HOEK, W. van der et al. (Ed.). Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010): Volume 1-3. Richland, SC: IFAAMAS, 2010. p. 807–814.
- FORNARA, N. et al. Artificial institutions: a model of institutional reality for open multiagent systems. Artificial Intelligence and Law, v. 16, n. 1, 2008.
- VIGANÒ, F.; COLOMBETTI, M. Model checking norms and sanctions in institutions. In: SICHMAN, J. S. a. et al. (Ed.). Coordination, Organizations, Institutions, and Norms in Agent Systems III. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, (Lecture Notes in Computer Science, v. 4870). p. 316–329.

References IX

CARDOSO, H. L.; OLIVEIRA, E. C. Institutional reality and norms: Specifying and monitoring agent organizations. Int. J. Cooperative Inf. Syst., v. 16, n. 1, p. 67–95, 2007.

BOISSIER, O. et al. Multi-agent oriented programming with jacamo. Science of Computer Programming, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, v. 78, n. 6, p. 747–761, 2013.