RBAC-MAS &
Agent Coordination Contexts

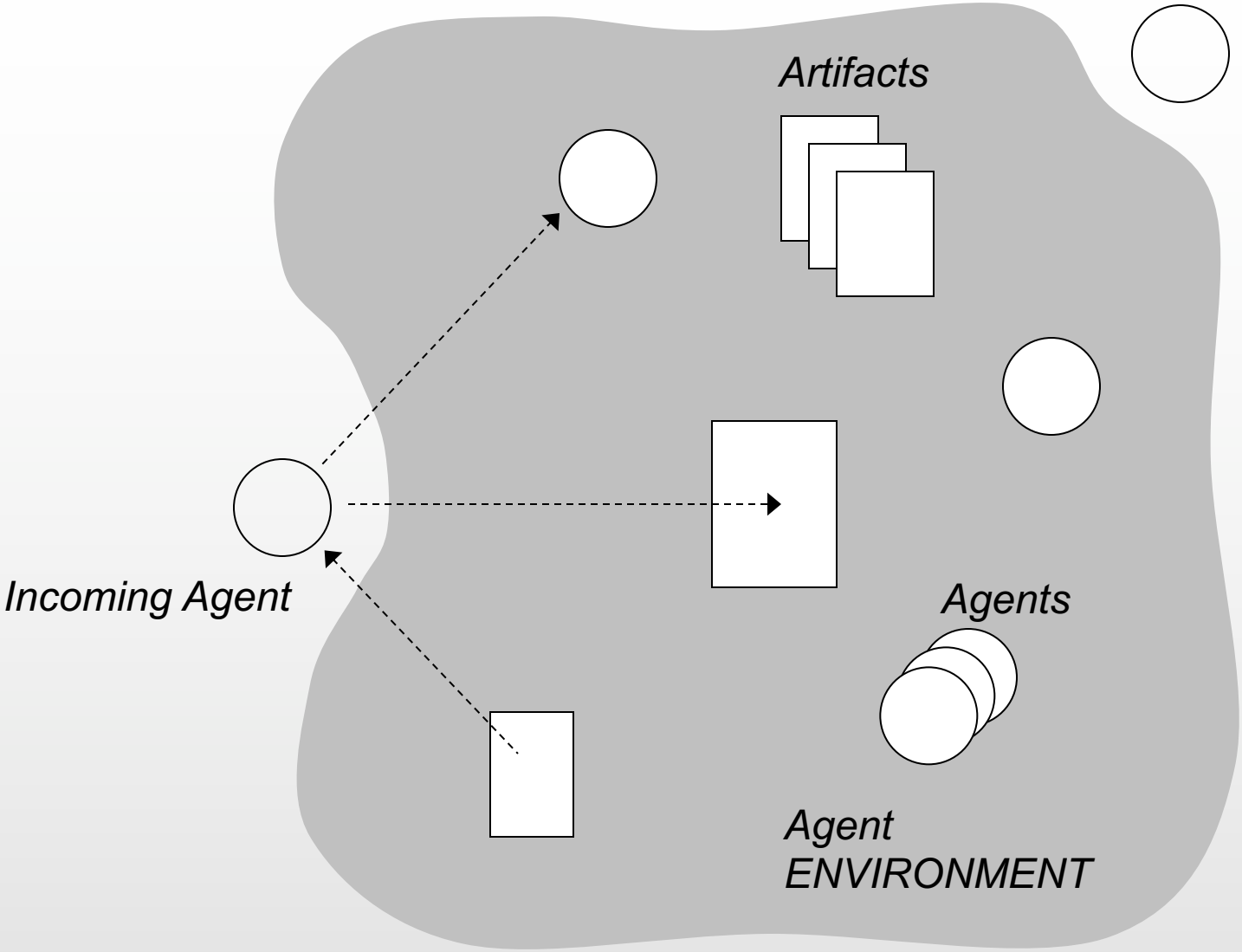
Andrea Omicini
after Mirko Viroli, Alessandro Ricci
Alma Mater Studiorum – Università di Bologna

MAS Scenario

- An open MAS where agents come in and interact with a (structured) environment
  - the agent workspace

  in order to achieve their own goals
- The individual view of the agent environment environment is made of
  - other agents
  - artifacts
    - modelling the agent environment
    - structuring the agent workspace
  - agents *speak* to agents, agents *use* artifacts, artifacts *link* with artifacts
    - [Omicini et al. @ AAMAS 2004 "Coordination Artifacts"],
    - [Viroli et al. @ E4MAS 2005 "Engineering the Environment with Artifacts"]

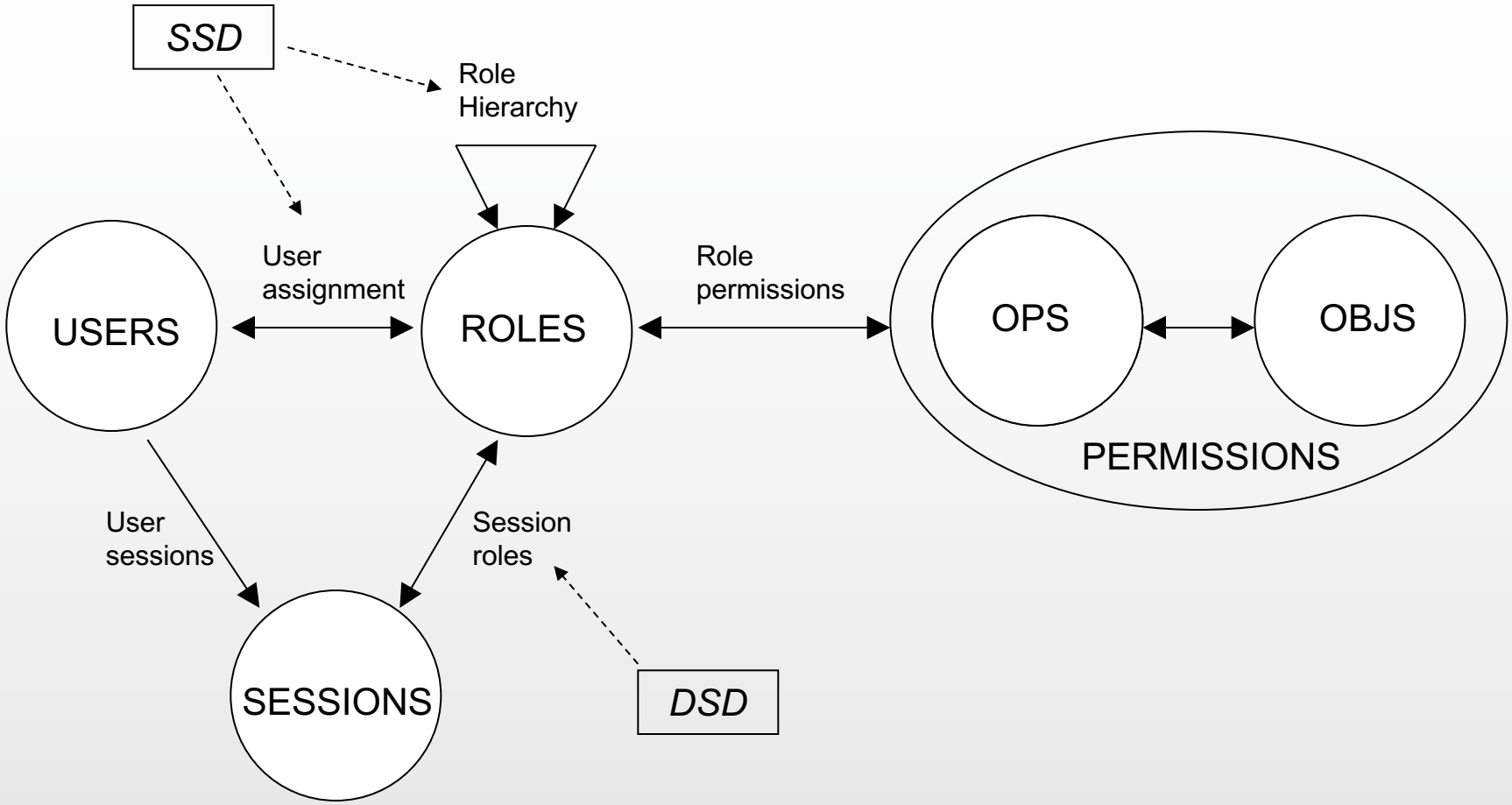
Reference MAS Architecture



Roles, RBAC & RBAC-MAS

- What are the consequences of this scenario on the management of a MAS organisation?
  - managing organisation as either a human or an agent
- In the most common and simplest acceptation, this means supporting a notion of *role*
  - each agent can enter a MAS only if authorised
  - each agent is allowed to play some given roles
  - each role enables / prevents given agent interactions within the MAS
    - hence supporting a notion of security as well
- In particular, which role model for a MAS?
  - we start from the RBAC model
  - and devise out our RBAC-MAS

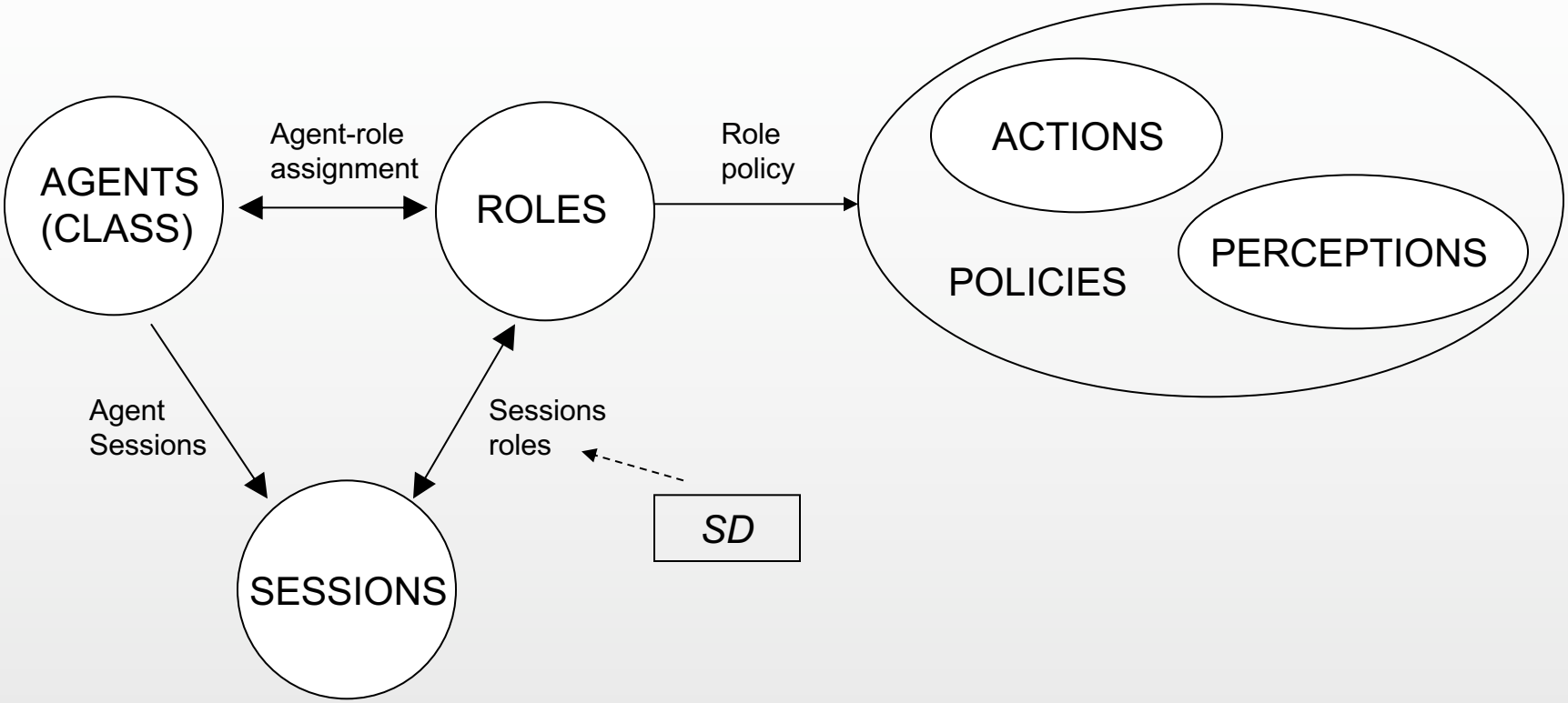
RBAC


SSD
Role
Hierarchy
User
assignment
Role
permissions
USERS
ROLES
OPS
OBJS
PERMISSIONS
User
sessions
Session
roles
SESSIONS
DSD

RBAC Features

- Role-Based Access Control (RBAC)
  - ["Proposed NIST standard for role-based access control", by Ferraiolo et al., ACM Transactions on Information and System Security, 2001]
  - makes it possible to establish relations
    - between roles
    - between permissions and roles
    - between users and roles
- The notion of role
  - is meant to model competency, authority and responsibility
  - is policy-neutral
- Interactions
  - are seen as operations over objects
  - generally model permissions in an abstract way
    - such as credit and debit
- Separation of Duties
  - rules for role inter-dependencies

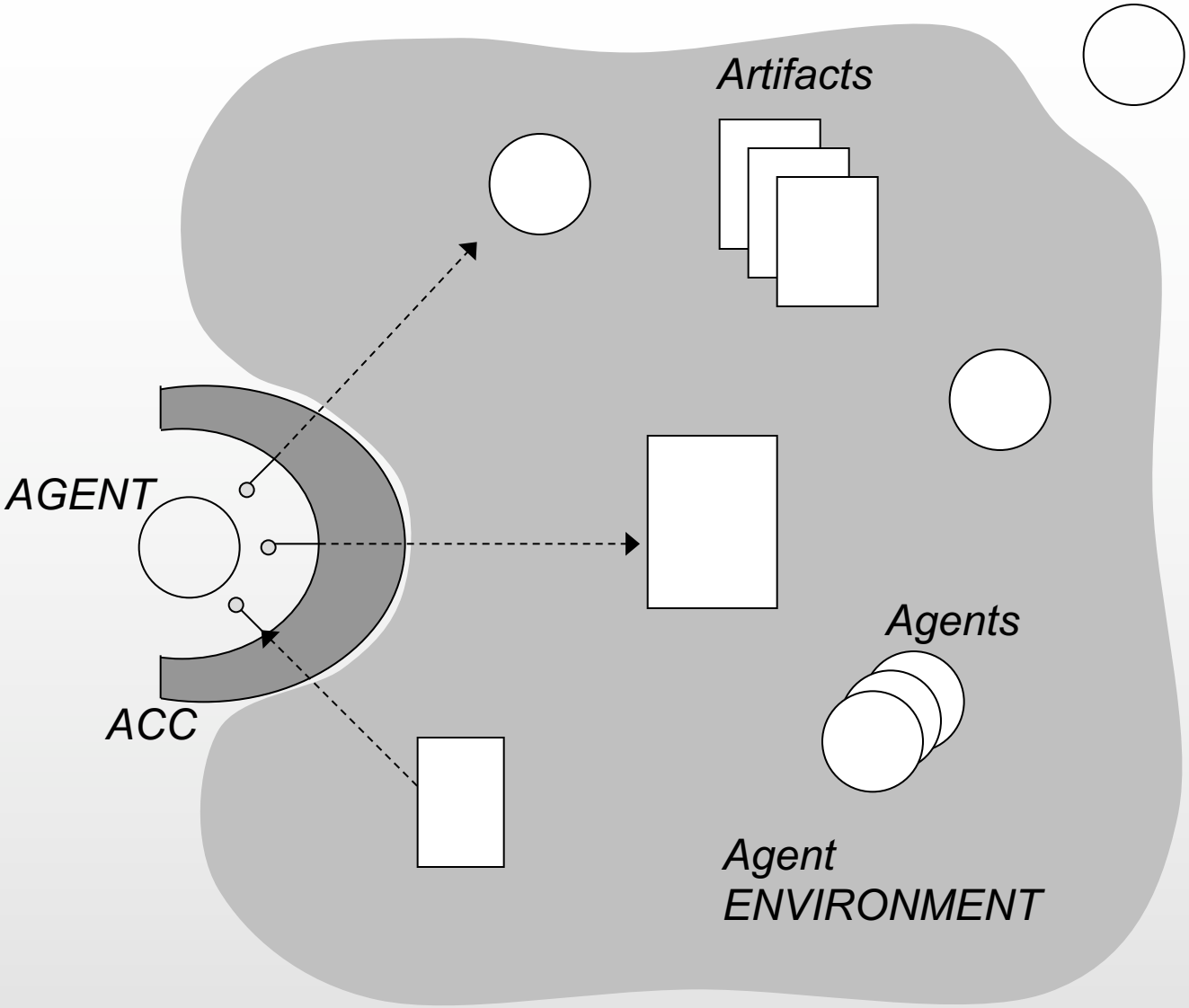
RBAC-MAS


AGENTS
(CLASS)
Agent-role
assignment
ROLES
Role
policy
ACTIONS
PERCEPTIONS
POLICIES
Agent
Sessions
Sessions
roles
SD
SESSIONS

RBAC-MAS Features

- RBAC-MAS
  - ["RBAC for Organisation and Security in an Agent Coordination Infrastructure", by Omicini&Ricci&Viroli, ENTCS 128(5)]
- Moving to MASs
  - *users* become *agent* (classes)
  - *operations* become *actions/perceptions*
  - *objects* become *agents/artifacts*
  - *permissions* become *policies* (protocol templates)
- How can we model sessions?
  - we need a run-time notion to enact this

Agent Coordination Context



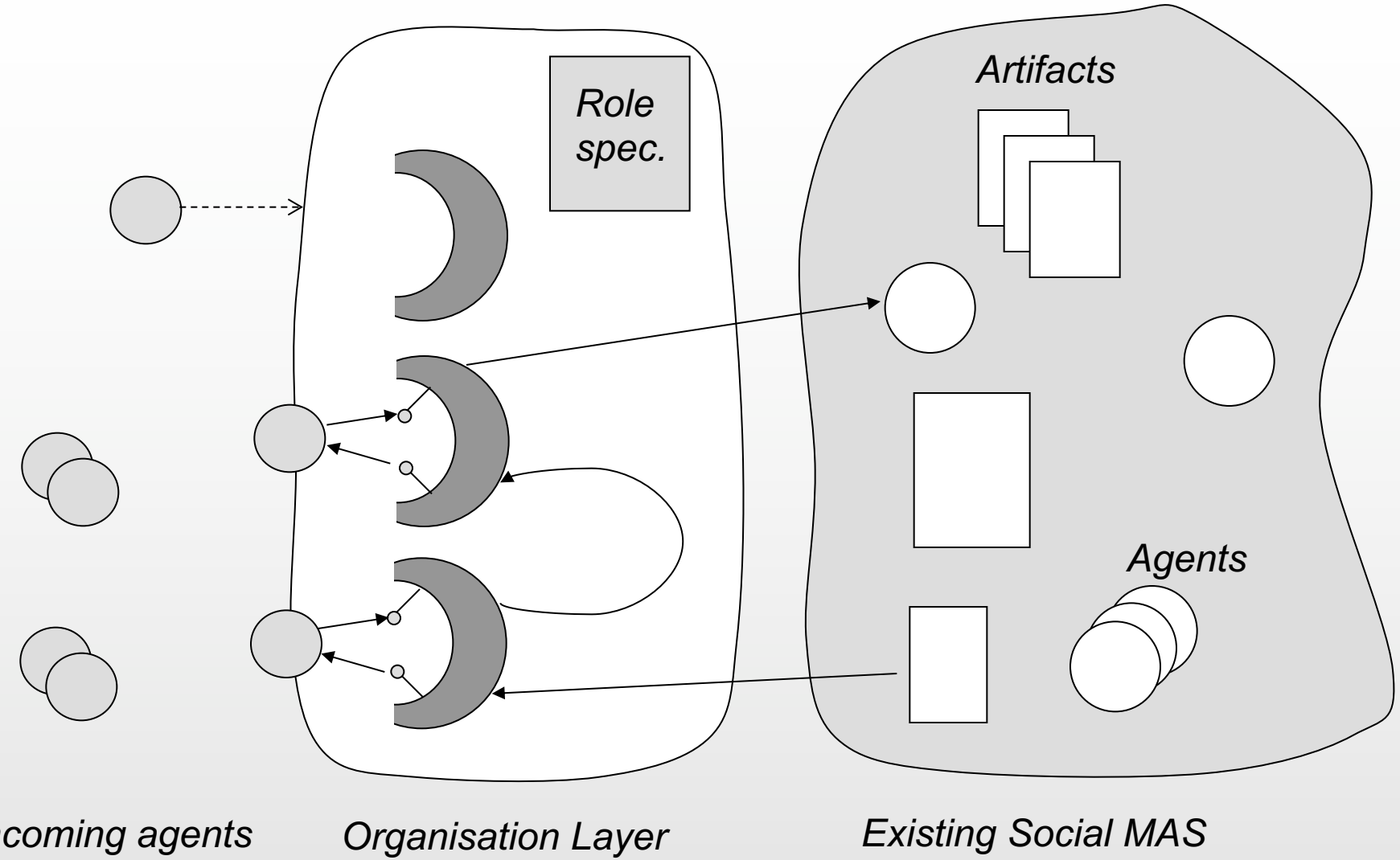
ACC Features

- Agent Coordination Contexts
  - ["Towards a Notion of Agent Coordination Context", by Omicini, in "Process Coordination and Ubiquitous Computing", CRC Press, 2002]
  - available in TuCSoN
    - http://tucson.sourceforge.net
- Represents and enacts
  - an interface for the agent towards the environment
  - the agent from the MAS viewpoint
- One ACC for each individual agent in the MAS
  - it is the only means by which an agent can interact with the MAS
  - the ACC enables only some given actions/perceptions
    - dynamically
    - according to a history-dependent policy

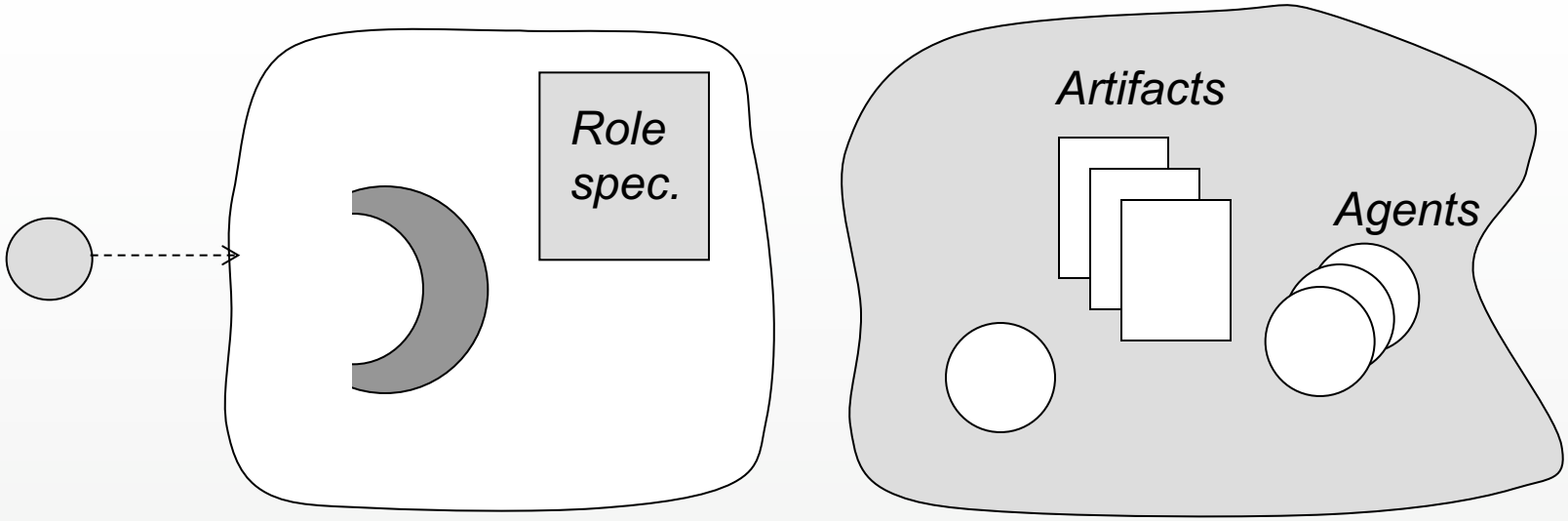
Organisation Infrastructure

- In principle, it an be added on top of any existing MAS infrastructure
  - it adopts the RBAC-MAS model
  - it uses ACCs to enact sessions
- MAS designer duties
  - role specification
    - that is, specify agent classes, roles, policies and their mutual relationships
- Infrastructure responsibilities
  - deliver ACCs to agents
  - maintain their global consistency
- Agent behaviour
  - is both *enabled* and *constrained* accordingly

Infrastructure View

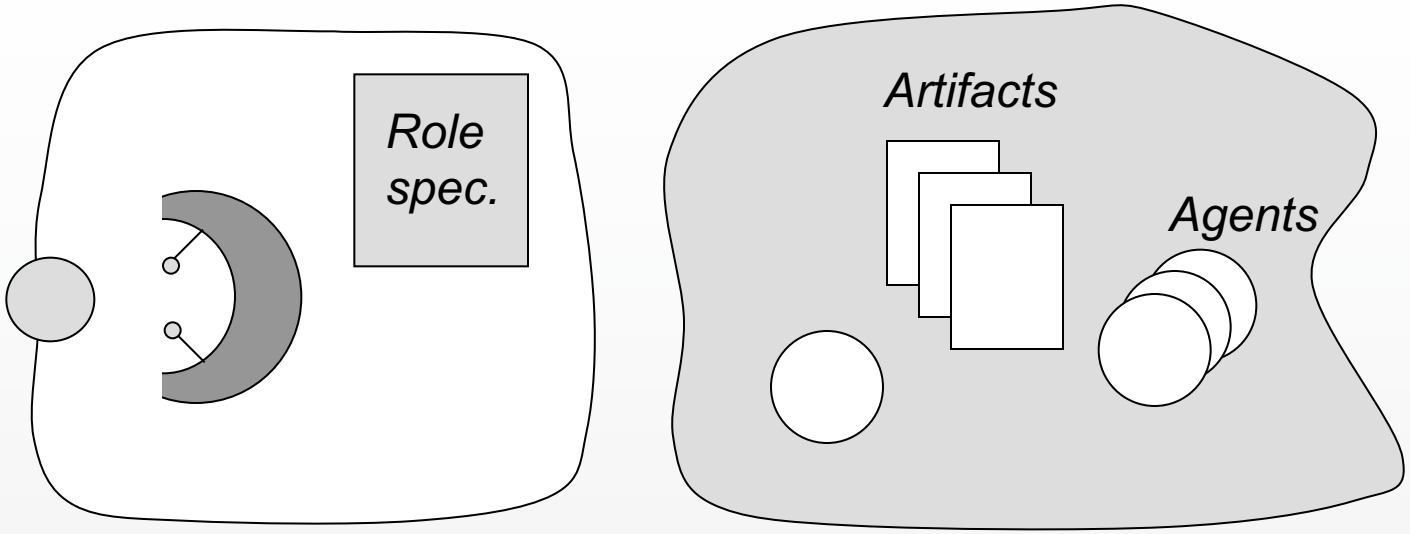


Phase: Entering a MAS



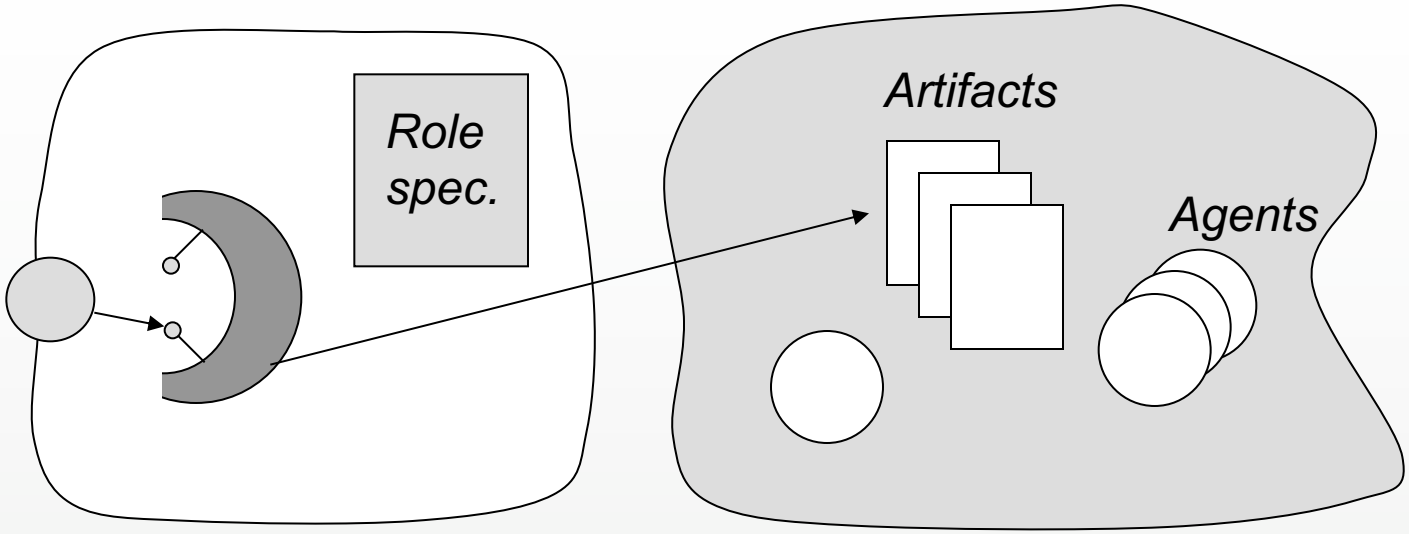
- The agent is first authenticated & classified
- The agent negotiates successfully a set of policies for interacting in the MAS
- A new ACC is prepared accordingly
- The agent enters the ACC

Phase: Role (De/)Activation



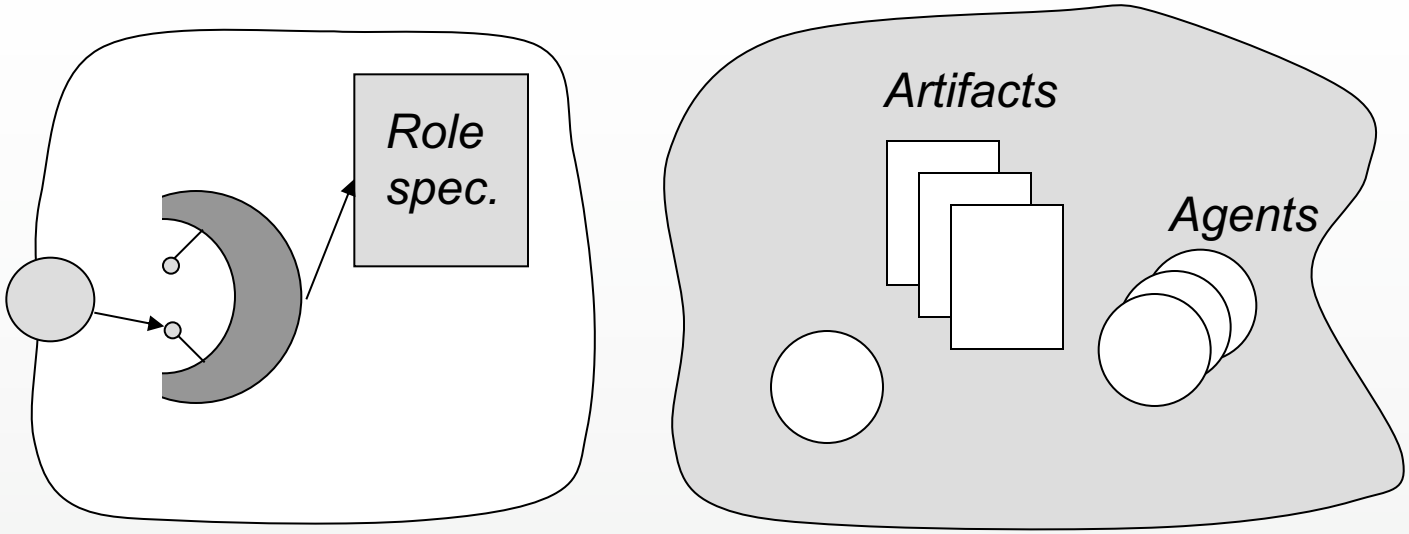
- The agent can ask for a role to be activated
- If this is granted, all the policies for that role are enabled by the ACC
- Eventually, a role can reach a state where it can be deactivated

Phase: Interacting in a MAS



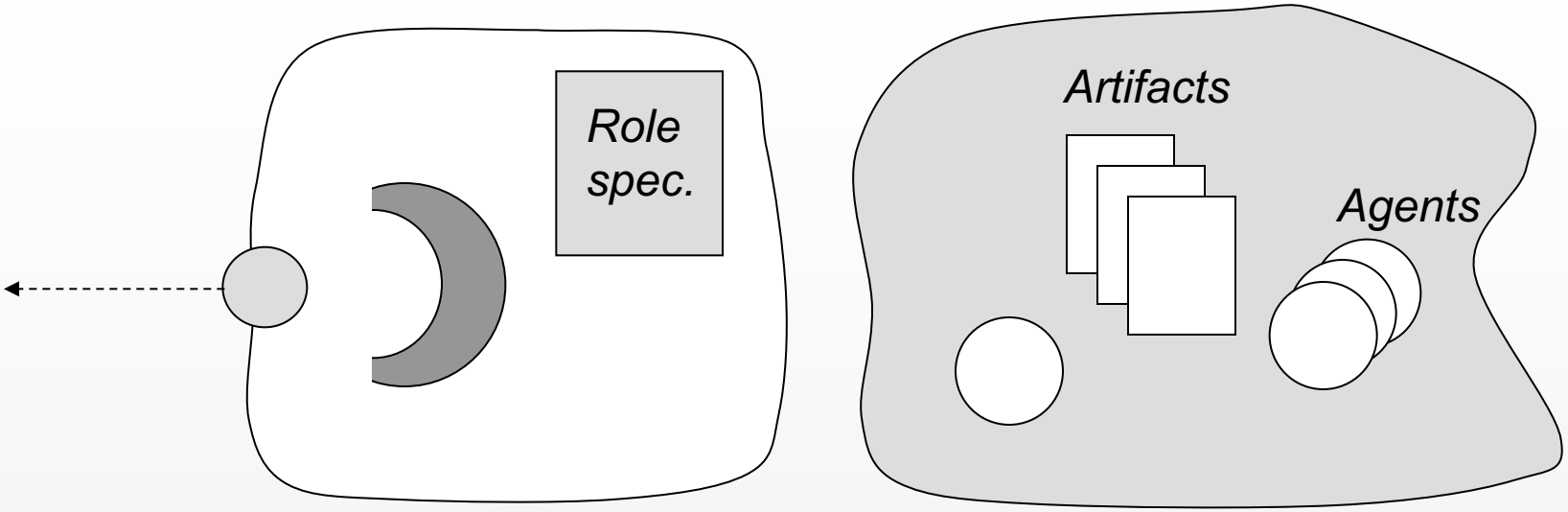
- Given a role and a policy of its, at a given time, the agent is enabled to execute one or more actions
  - this information can be inspected by the agent
- When an action is executed, and the state of the policy state is accordingly changed in a dynamic way
  - hence a policy can specify true protocols

Phase: Managing Organisation



- Some roles may even enable agents to manage the organisation
  - reading/adding/removing/updating roles, policies, …

Phase: Quitting MAS



- If no roles are currently activated, the agent can ask to leave the ACC, hence the society

Infrastructure Design

- Formal specification based on process algebras
  - ["An Algebraic Approach for Modelling Organisation, Roles and Contexts in MAS", by Omicini&Ricci&Viroli, AAECC Journal 2005]
- Syntax of a whole MAS configuration
  - specified with a BNF grammar
- Dynamic aspects of MAS
  - managed by a layered operational semantics
  - for agents, ACCs, negotiation aspects

Syntax

S	$::=$	$X \| O \| E$	MAS configuration
X	$::=$	0	agent configuration
	$\|$	$\langle a, A \rangle$	inert agent
	$\|$	$\langle a, (C)A \rangle$	active agent
	$\|$	$X \| X$	agents
O	$::=$	$K \| V$	organisation configuration
K	$::=$	0	structure configuration
	$\|$	$\{c(n)\}_{\mathbb{C}}$	agent class (with cardinality)
	$\|$	$\{c, r\}_{\mathbb{CR}}$	class role
	$\|$	$\{r, p\}_{\mathbb{RP}}$	role policy
	$\|$	$\{p := P\}_{\mathbb{P}}$	policy definition
	$\|$	$\{[RS] + r\}_{\mathbb{SD}}$	separation of duty constraint
	$\|$	$K \| K$	control structures
RS	$::=$	$0 \mid r \mid RS \| RS$	roleset
V	$::=$	0	activity configuration
	$\|$	$[a, c]_{\mathbb{A}}$	active agent class
	$\|$	$[a, r]_{\mathbb{AR}}$	active agent role
	$\|$	$V \| V$	activity controls

Agent Behaviour and Actions

$A ::= 0 \mid \alpha.A \mid A + A$	agent action configuration
$\alpha ::= r : p : \phi \mid \nu$	agent actions
$\phi ::= \epsilon \mid \omega$	environment / organisation actions
$\omega ::=$	organisation actions
$+K$	control structure addition
$\mid -K$	control structure removal
$\mid K \mapsto K$	control structure update
$\mid ?K$	control structure reading
$\nu ::=$	ACC actions
$\downarrow$	ACC entry
$\mid \uparrow$	ACC exit
$\mid +r : R$	role activation
$\mid -r$	role deactivation
$\iota ::= a\nu$	negotiation actions

ACC Configuration

C	$::= 0$	ACC configuration
	$\mid r : R$	active role
	$\mid C \| C$	ACCs
R	$::= 0$	role
	$\mid p : P$	active policy
	$\mid R \| R$	roles
P	$::= 0$	policy
	$\mid \pi$	controlled action
	$\mid P; P$	sequence
	$\mid P + P$	choice
	$\mid P \| P$	interleaving
	$\mid \mathcal{D}$	(recursive) definition call
π	$::= \phi \mid \diamond$	policy actions

Interaction Rule

$$\frac{A \xrightarrow{r:p:\sigma\phi}_{\mathbb{A}} A' \qquad P \xrightarrow{\sigma \rhd \phi}_{\mathbb{P}} P'}{\langle a, (C \| r : (R \| p : P)) A \rangle \xrightarrow{\sigma \rhd \phi}_{\mathbb{X}} \langle a, (C \| r : (R \| p : P')) A' \rangle} \text{[ACT]}$$

- Agent *A* in an ACC with role *R* and policy *P*
- Executes action ϕ , subject to substitution σ
- The ACC policy accordingly moves to *P′*

Operational Rules

$$\frac{A \xrightarrow{\downarrow}_{\mathbb{A}} A'}{\langle a, A\rangle \xrightarrow{a\downarrow}_{\mathbb{X}} \langle a, (0)A'\rangle} \quad \text{[ENTER]}$$

$$\frac{A \xrightarrow{\uparrow}_{\mathbb{A}} A'}{\langle a, (0)A\rangle \xrightarrow{a\uparrow}_{\mathbb{X}} \langle a, A'\rangle} \quad \text{[QUIT]}$$

$$\frac{A \xrightarrow{+r:R}_{\mathbb{A}} A'}{\langle a, (C)A\rangle \xrightarrow{a+r:R}_{\mathbb{X}} \langle a, (C \| r : R)A'\rangle} \quad \text{[ACTIVATE]}$$

$$\frac{\begin{array}{c}\langle a, (C \| r : R)A\rangle \xrightarrow{a-r}_{\mathbb{X}} \langle a, (C)A'\rangle \\ \left(P \xrightarrow{\sigma \triangleright \diamond}_{\mathbb{P}} P' \quad \text{or} \quad P \not\rightarrow_{\mathbb{P}}\right)\end{array}}{\langle a, (C \| r : (R \| p : P))A\rangle \xrightarrow{a-r}_{\mathbb{X}} \langle a, (C)A'\rangle} \quad \text{[DEACT-REC]}$$

$$\frac{A \xrightarrow{-r}_{\mathbb{A}} A'}{\langle a, (C \| r : 0)A\rangle \xrightarrow{a-r}_{\mathbb{X}} \langle a, (C)A'\rangle} \quad \text{[DEACT-FIX]}$$